AFRL-VA-WP-TR-1999-3050

DEVELOPMENT OF THE AERODYNAMIC/AEROSERVOELASTIC MODULES IN ASTROS

VOLUME 2: ZAERO PROGRAMMER'S MANUAL (F33615-96-C-3217)

P.C. CHEN
D. SARHADDI
D.D. LIU

ZONA Technology, Inc.
7430 E. Stetson Drive, Ste 205
Scottsdale, AZ 85251

FEBRUARY 1999

FINAL REPORT FOR PERIOD SEPTEMBER 1996 – SEPTEMBER 1998

Approved for public release; distribution unlimited

AIR VEHICLES DIRECTORATE
AIR FORCE RESEARCH LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7542

NOTICE

USING GOVERNMENT DRAWINGS, SPECIFICATIONS, OR OTHER DATA INCLUDED IN THIS DOCUMENT FOR ANY PURPOSE OTHER THAN GOVERNMENT PROCUREMENT DOES NOT IN ANY WAY OBLIGATE THE UNITED STATES GOVERNMENT. THE FACT THAT THE GOVERNMENT FORMULATED OR SUPPLIED THE DRAWINGS, SPECIFICATIONS, OR OTHER DATA DOES NOT LICENSE THE HOLDER OR ANY OTHER PERSON OR CORPORATION; OR CONVEY ANY RIGHTS OR PERMISSION TO MANUFACTURE, USE, OR SELL ANY PATENTED INVENTION THAT MAY BR RELATED TO THEM.

THIS REPORT IS RELEASEABLE TO THE NATIONAL TECHNICAL INFORMATION SERVICE (NTIS). AT NTIS, IT WILL BE AVAILABLE TO THE GENERAL PUBLIC, INCLUDING FOREIGN NATIONS.

THIS TECHNICAL REPORT HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION.

VICTORIA A. TISCHLER
Aerospace Engineer
Design and Analysis Branch

VIPPERLA B. VENKAYYA
Leader, Multidisciplinary Design
Design & Analysis Branch

NELSON D. WOLF, Chief
Design and Analysis Branch
Structures Division

IF YOUR ADDRESS HAS CHANGED, IF YOU WISH TO BE REMOVED FROM OUR MAILING LIST, OR IF THE ADDRESSEE IS NO LONGER EMPLOYED BY YOUR ORGANIZATION, PLEASE NOTIFY AFRL/VASD BLDG 45, 2130 8TH STREET, SUITE 1, WRIGHT-PATTERSON AFB OH 45433-7542 TO HELP MAINTAIN A CURRENT MAILING LIST.

COPIES OF THIS REPORT SHOULD NOT BE RETURNED UNLESS RETURN IS REQUIRED BY SECURITY CONSIDERATIONS, CONTRACTUAL OBLIGATIONS, OR NOTICE ON A SPECIFIED DOCUMENT.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave Blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	FEBRUARY 04, 1999	FINAL 24 SEP 1996 – 24 SEP 1998

4. TITLE AND SUBTITLE
DEVELOPMENT OF THE AERODYNAMIC/AEROSERVOELASTIC MODULES IN ASTROS
VOLUME 2 – ZAERO PROGRAMMER'S MANUAL

5. FUNDING NUMBERS

C: F09603-95-D-0175
PE: 65520F
PR: STTR
TA: 41
WU: 00

6. AUTHOR(S)
P. C. Chen, D. D. Liu, D. Sarhaddi
ZONA Technology, Inc.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
ZONA Technology, Inc.
7434 E. Stetson Drive, Suite 205
Scottsdale, AZ 85251
Tel 602-945-9988 / Fax 602-945-6588

8. PERFORMING ORGANIZATION REPORT NUMBER

ZONA 99-11B

9. SPONSORING/MONITORING AGENCY(S) AND ADDRESS(ES)
Air Vehicles Directorate
Air Force Research Laboratory
Air Force Materiel Command
Wright-Patterson Air Force Base, Oh 45433-7542
POC: Dr V. B. Venkayya, AFRL/VASD, 937-255-2582

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

AFRL-VA-WP-TR-1999-3050

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)

This report is a part of the documentations which describe the complete development of an STTR Phase II effort entitled, "Development of the Aerodynamic/Aeroservoelastic Modules in ASTROS." This report is one of four manuals that comprise the final report. The remaining reports consist of the ZAERO User's Manual (Volume I), the ZAERO Applications Manual (Volume III) and the ZAERO Theoretical Manual (Volume IV).

ASTROS* is the seamless integration of the ZAERO module into ASTROS. As an aerodynamic enhancement to ASTROS, ZAERO is the ZONA aerodynamic module, unified for all Mach number ranges. This manual assumes the reader is familiar with the ASTROS system architecture, terminology and programming environment. In particular, it is geared toward system administrators and/or programmers working within the ASTROS* environment.

First, an overview of ZAERO and ASTROS* is presented. The modified system generation (SYSGEN) input for ASTROS* accommodating the ZAERO module is presented next, along with an ASTROS* system generation flow chart. Third, nine ZAERO engineering application modules within the ASTROS* environment are described. Lastly, the ZAERO specific relational and matrix database entity descriptions are presented.

14. SUBJECT TERMS
Multidisciplinary Optimization, ZAERO Module, ASTROS*, Subsonic-Transonic-Supersonic-Hypersonic Aerodynamics, Aeroelasity, Aeroservoelasticity, Flutter

15. NUMBER OF PAGES
103

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF THIS ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR

Table of Contents

List of Figures

List of Tables

FOREWORD

This interim report is submitted in fulfillment of CDRL CLIN 0001, Data Item A009, Title: Interface Design Document of a Small Business Technology Transfer (STTR) contract No. Contract No. F33615-96-C-3217 entitled, "Development of the Aerodynamic/Aeroservoelastic Modules in ASTROS," covering the performance period from 24 September 1996 to 24 September 1998. This document provides the programmer's documentation for the ZAERO module in ASTROS*.

This work was performed by ZONA Technology, Inc. and its subcontractors, the University of Oklahoma (Research Institute)/Technion (I.I.T) and Universal Analytics Inc. This work is the second phase of a continuing two-phase STTR contract supported by AFRL/Wright-Patterson. The first phase STTR contract No. F33615-95-C-3219 entitled, "Enhancement of the Aeroservoelastic Capability in ASTROS," was completed in May 1996 and published as WL-TR-96-3119. Started in September 1996, the present second phase STTR contract was conducted by the same team members as in phase I. These contributors are: P.C. Chen (P.I.), D. Sarhaddi and D.D. Liu of ZONA Technology Inc.; Fred Striz of the University of Oklahoma; Moti Karpel of Technion/I.I.T.; and Tony Shimko and Steve Chen of Universal Analytics.

This STTR contract is sponsored by AFRL/Wright-Patterson. Capt. Gerald Andersen is the contract monitor and Dr. V.B. Venkayya is the initiator of the whole STTR effort. During the course of the present phase on the development of ASTROS*, the technical advice and assistance received from Mr. Doug Neill of The MacNeal Schwendler Corporation, Dr. V.B. Venkayya and others from AFRL are gratefully acknowledged.

1.0 INTRODUCTION

There are four major documents that describe the ZONA Aerodynamics Module (ZAERO) Module which has been seamless integrated into the Automated STRuctural Optimization System (ASTROS). These are: the ZAERO User's, Programmer's, Application and Theoretical Manuals for ASTROS*. While ZAERO represents the ZONA Aerodynamics Module, ASTROS* is defined as the seamless integration of ZAERO into ASTROS, i.e. ASTROS* = ZAERO + ASTROS. This Programmer's Manual gives the detailed description of the ZAERO software and its interface with the ASTROS system. Newly created database entities in support of the ZAERO module within ASTROS* are described. Newly developed engineering application modules comprising the ZAERO module are presented in detail.

This manual assumes that the user is familiar with the ASTROS system (Version 11.0), its terminology and programming environment. A complete and comprehensive description of the ASTROS environment can be found in the ASTROS User's and Programmer's Manuals (Refs 1,2). In particular, this manual is geared toward system administrators and/or programmers within the ASTROS* enviroment.

Section 2 presents an overview of the ZAERO software, its aerodynamic capability over that of the previous modules in ASTROS, and the program architecture of ZAERO in relation to ASTROS.

Section 3 presents the computer files delivered under this contract which contain all of the subroutines of the ZAERO module, the modified System Generation (SYSGEN) input for ASTROS*, and the ASTROS* system generation process.

Section 4 presents the ZAERO engineering application modules (altogether nine modules) that make up ZAERO within the ASTROS* environment. Together with the ASTROS* object library, these ZAERO engineering applications modules constitute the entire ASTROS* executable (see ASTROS* system generation flow chart).

Section 5 presents the ZAERO specific relational and matrix database entity descriptions established upon building of the ASTROS* system that are used for communication of data among the ZAERO engineering application modules.

2.0 ZAERO MODULE AND ASTROS*

ASTROS (Automated STRuctural Optimization System) is a finite element based procedure tailored for the preliminary design of aerospace structures. As such, it includes flexibility and generality in multiple discipline integration. For aircraft, missile or spacecraft design, the unique attributes of ASTROS lie in its savings of design effort and time, improvement in flight performance and reduction in structural weight. In principle, ASTROS was aimed at the effective multidisciplinary interactions between aerodynamics, aeroelastics, structures and other modules. Although today a well-aclaimed, proven tool for Multidisciplinary Optimization (MDO) and analysis, ASTROS still requires further improvement in its capabilities in steady/unsteady aerodynamics, aeroelasticity and aeroservoelasticity (e.g. Ref 3).

The ZONA aerodynamic codes contained in the ZAERO module are the software products of ZONA Technology developed throughout the years. These include four major steady/unsteady aerodynamics codes, namely ZONA6, ZONA7, ZTAIC, and ZONA7U, that jointly cover the complete domain of all Mach number ranges. The ZONA aerodymanic system (the ZAERO System) which contains the ZAERO module and two other modules were developed under the support of AFRL/Wright-Patterson AFB for their seamless integration into the ASTROS system to improve and enhance the capability of ASTROS in aerodynamics, aeroelasticity and aeroservoelasticity (ASE). In particular, the ZAERO module improves the aerodynamics capability over the earlier aerodynamics modules in ASTROS in the following aspects (also see Figs 1 and 2):

1. Wing-Body geometry input for realistic aircraft configurations including external stores.
2. Flight regimes that include subsonic, supersonic, transonic and hypersonic Mach numbers.
3. High-order paneling scheme to assure accurate and robust solutions (without stringent paneling requirements).
4. Provides Aerodynamic Influence Coefficient (AIC) matricies for all flow regimes including the generation of transonic AIC.
5. Steady/unsteady aerodynamic options for static and dynamic aeroelastic applications.
6. Unified aerodynamic geometry bulk data input.

The development and seamless integration of the the ZAERO System into ASTROS has created a unique Multidisciplinary Design/Analysis and Optimization (MDO/MAO) tool that is currently unsurpassed in its steady/unsteady aerodynamic and aeroelastic capability. The ZAERO System consists of essentially three modules which include the ZAERO module, the AGM (aerodynamic geometry module) and the 3D-Spline module (see Fig 3).

As can be seen in Fig 1, current capabilities of ASTROS and NASTRAN are limited to subsonic and supersonic Mach numbers and applicable to lifting surfaces only. By contrast, ZAERO is valid throughout the full range of subsonic to hypersonic Mach numbers and is applicable to complex aircraft configurations with external stores.

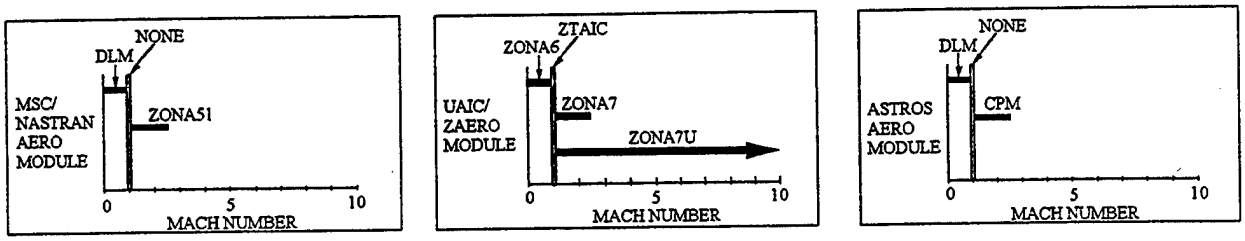

Figure 1. ZAERO and Other Aerodynamic Modules.

Fig 2 shows the capability of each code in the ZAERO Module (marked with †) along with other ZONA Codes.

Capability		ZONA Unsteady/Steady Aerodynamic Codes – ZAERO						
		ZONA51	*ZONA51U*	*ZONA7*†	*ZONA7U*†	*ZONA6*†	*ZTAIC*†	*ZTAIC6*
Geometry	• Lifting Surface (L.S.)	●	●	●	●	●	●	●
	• Thickness Effect		●		●		●	●
	• L.S. + Body = Whole Aircraft			●	●	●		●
Mach Number	• Subsonic					●	●	●
	• Transonic						●	●
	• Supersonic	●	●	●	●			
	• Hypersonic		●		●			

Figure 2. Capability of the ZAERO Module.

The seamlessly integrated ZAERO System in ASTROS is called ASTROS*. Fig 3 illustrates the role of the ZAERO System within ASTROS* and the overall ASTROS* program architecture. The ZAERO System consists of three primary modules with the following functionalities:

- *Unified Aerodynamic Geometry Module (AGM)*
  The Unified Aerodynamic Geometry Module processes the ZAERO model aerodynamic geometry input. Two newly created bulk data entries are used to define the aerodynamic geometry, namely **CAERO7** for wing-like components such as wings, tails, pylons, launchers and store fins, and **BODY7** for body-like components such as fuselage, stores and missile bodies.

- *3-D Spline Module*
  The 3-D Spline Module provides for the interconnection between the aerodynamic and structural models through the generation of spline matricies. Three spline methods are supported by this module. These are the infinite plate spline (IPS) method (**SPLINE 1**), the beam spline method (**SPLINE 2**) and the thin plate spline (TPS) method (**SPLINE 3**). The TPS

is an addition to the spline capability provided by ASTROS and unlike the IPS method does not require that a spline plane be defined.

- *The ZAERO Module*
  The ZAERO Module is made up of the four major aerodynamic codes (ZONA6, ZONA7, ZTAIC, ZONA7U) and generates the Unified Aerodynamic Influence Coefficient (UAIC) matrices, gust force vectors, control surface aerodynamic vectors and steady aerodynamic force vecotrs of trim parameters.

Database entities generated by AGM, 3-D Spline and ZAERO modules are computed in the ASTROS* preface phase and are not recomputed in the analysis/optimization loop.

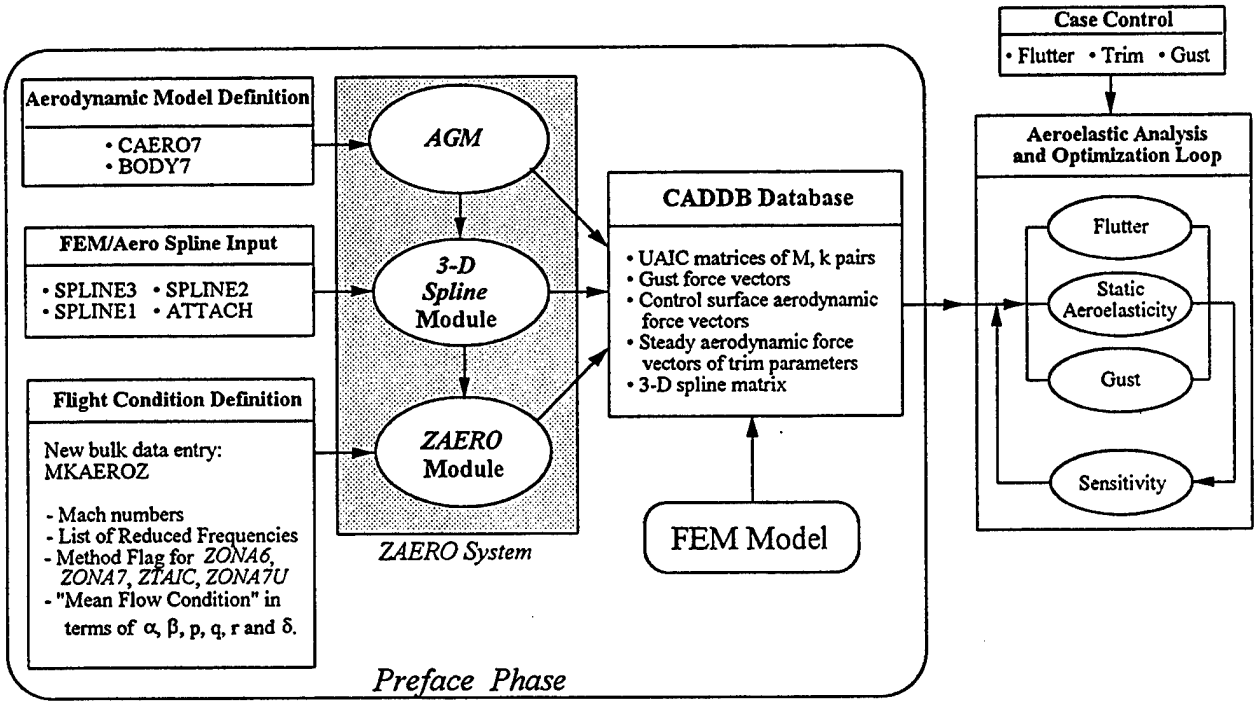

Figure 3. ASTROS/ZAERO (ASTROS*) Program Architecture.

3.0 ASTROS* SYSTEM GENERATION

3.1 Generation of the ASTROS* System

The ASTROS System Generation Process (SYSGEN) has been modified to include the compilation of the ZAERO module source code and the linking of the ZAERO module object code into the ASTROS system. For ease of use, the system generation process has been kept the same as that of ASTROS (Version 11.0). The change made to this process to incorporate the ZAERO module are:

1. Updates to the SYSGEN input files (described in Sections 3.2.1 through 3.2.5)
2. Modified script file **Makexqdriv** for compiling the ZAERO module source code (described in Section 3.1.1)
3. Modified script file **Makeastros** for linking of the ZAERO module object code into the ASTROS* system (described in Section 3.1.2)

The entire SYSGEN process is depicted in Figure 4 and is briefly outlined as follows.

The modified SYSGEN input files (1) are processed by SYSGEN (2). SYSGEN generates the ASTROS* System Database (SYSDB) (3), SYSGEN output file (4) and the fortran source code XQDRIV (5). Both the ZAERO engineering applications modules (6) and XQDRIV source code (5) are compiled by the **Makexqdriv** script file (7). The object library of ASTROS (Version 11.0) (8) and object files generated by **Makexqdriv** (7) are linked via the **Makeastros** script file called by **astlink** (9) to generate the ASTROS* Executable Image (10). The ASTROS* System Database (3) and ASTROS* Executable (10) make up the ASTROS* system.

3.1.1 Compiling the ZAERO Module

The Makefile (**Makexqdriv**) used to compile the **XQDRIV** file generated by SYSGEN and located in the ASTROS (Version 11.0) sysgen directory has been updated to compile the ZAERO source files listed in Table 1 (see Figure 5). Should any modifications to the source code be required, the corresponding files where changes are made must be re-compiled in **Makexqdriv**. If no changes are made and the user wishes to re-build the ASTROS* system, it is not necessary to re-compile these files. Therefore all corresponding lines in **Makexqdriv** can be commented out to speed up the ASTROS* regeneration process.

3.1.2 Linking the ZAERO Module

The Makefile (**Makeastros**) called by the **astlink** script file to relink ASTROS* and located in the ASTROS (Version 11.0) sysgen directory has been updated to link the ZAERO object files generated upon the compilation in **Makexqdriv** (see Figure 6).

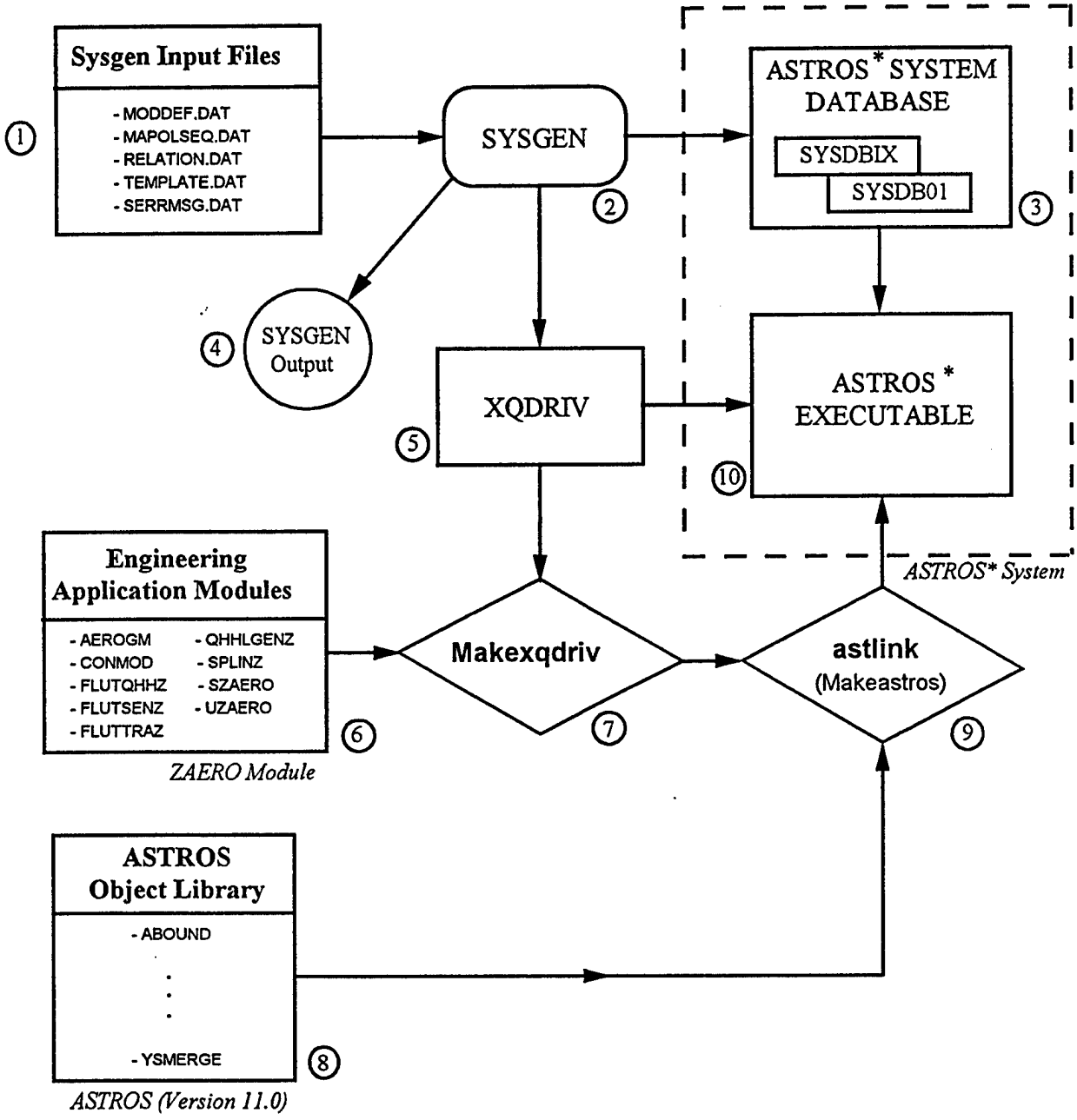

Figure 4. ASTROS* System Generation Process.

```
update: compflgs

# clean up

        @ rm xqdriv.o
        @echo "astros.a now up to date"
xqdriv.o:  xqdriv.f
        @echo ""
        @echo "compiling xqdriv.f with the " \"$(FC)\" " compiler and flags " \"$(FFLAGS)\"
        @echo ""
        $(FC) $(FFLAGS) -c xqdriv.f
#
# ZAERO Source Files
# ----------------------------------
        @echo ""
        $(FC) $(FFLAGS) -c aerogm.f
        @echo ""
        $(FC) $(FFLAGS) -c fltqhz.f
        @echo ""
        $(FC) $(FFLAGS) -c splinz.f
        @echo ""
        $(FC) $(FFLAGS) -c utility.f
        @echo ""
        $(FC) $(FFLAGS) -c zaerom.f
# ----------------------------------
#
        @echo ""
        $(FC) $(FFLAGS) -c XXBD.f
# now update the astros library with the new xqdirv
        @echo ""
        @echo "updating astros.a ... "
        /usr/ccs/bin/ar $(ARIFLGS) astros.a xqdriv.o
compflgs:
        @/usr/ccs/bin/make -f Makeflags $(TARGET) "MFILE = Makexqdriv" "RETURN = xqdriv.o"
```

THIS SECTION CAN BE COMMENTED OUT WITH (#) IF NO CHANGES ARE MADE TO THE ZAERO SOURCE CODE

Figure 5. Modified Makexqdriv File for ASTROS*.

```
# procedure to relink astros

#

# get the fortran compiler flags from Makeflags
#
load:  compflgs
        @echo " "
        @echo "Linking complete new version of astros now exsits"
compflgs:
        @/usr/ccs/bin/make -s -f Makeflags $(TARGET) "RETURN=lastros" "MFILE=Makeastros"
lastros:
        @echo""
        @echo "Generating a new version of astros.bin"
        @echo ""
        @/usr/ccs/bin/ar $(AROFLGS) astros.a astros.o
        @echo "Relinking astros .... ( This will take a few minutes )"
#       @$(LINK) $(ENTRY) $(LINIT) -o astros.bin astros.o aerogm.o XXBD.o astros.a $(LIBS)
        @$(LINK) $(ENTRY) $(LINIT) -o astros.bin  astros.o XXBD.o aerogm.o fltqhz.o splinz.o utility.o
zaerom.o zaerolib.o astros.a $(LIBS)
        @rm astros.o
```

Figure 6. Modified Makeastros File for ASTROS*.

3.2 ZAERO Sysgen Input

To facilitate the ASTROS* system generation described in Section 3.1, the five SYSGEN input data files, namely **MODDEF.DAT**, **MAPOLSEQ.DAT**, **TEMPLATE.DAT**, **RELATION.DAT** and **SERRMSG.DAT**, have been modified to include all components necessary for integration of ZAERO in ASTROS*. Modifications to each of these files are described in the following subsections. The physical changes made to each of these files are presented in Appendicies A through E, respectively.

3.2.1 Functional Module Definintion (MODDEF.DAT)

The ASTROS* run-time library of MAPOL addressable modules file (**MODDEF.DAT**) has been updated to account for all newly developed engineering application modules presented in Section 5. These module definitions provide the additional links between the ASTROS* executive system and the ZAERO engineering application modules. The ZAERO functional module defintions are presented in Appendix A. For a detailed description of this file, please see Ref 2.

3.2.2 MAPOL Sequence (MAPOLSEQ.DAT)

For seamless integration of ZAERO into ASTROS, the ASTROS MAPOL sequence (file **MAPOLSEQ.DAT**) has been modified. The complete ASTROS* MAPOL sequence listing is presented in Appendix B. All changes to the original ASTROS (Version 11.0) MAPOL sequence listing are highlighted in boldface text and are demarcated by arrows on the right. For a detailed description of this file, please see Ref 2.

3.2.3 Bulk Data Template Definition (TEMPLATE.DAT)

In the development of the ZAERO module, twenty three new bulk data entries were created. Bulk data template definitions for these new bulk data entries were added to those of ASTROS (Version 11.0) and are presented in Appendix C. For a detailed description of this file, please see Ref 2.

3.2.4 Relational Schema Definition (RELATION.DAT)

Schema definitions of all relational database entities used by the ZAERO module have been defined in file RELATION.DAT. These relational entity schema definitons are presented in Appendix D. For a detailed description of this file, please see Ref 2.

3.2.5 Error Message Text Definition (SERRMSG.DAT)

Three new error message definition modules have been developed corresponding to the following engineering application modules: AEROGM, SPLINZ and ZAEROM. These ZAERO error message module definitions are presented in Appendix E. For a detailed description of this file, please see Ref 2.

3.3 The ZAERO Software

Under the current contract, six computer files containing all ZAERO engineering application and utility modules are delivered. These six files along with corresponding file descriptions are listed in Table 1. These files contain all of the ZAERO engineering application modules.

Table 1. Computer Files Comprising ZAERO.

File Name	Description	File Type
aerogm.f	Code for processing of the wing/body aerodynamc geometry used by all ZAERO aerodynamic methods	source
fltqhz.f	Code for processing of matrices required for flutter analysis, including a newly developed K-method	source
splinz.f	Code for processing of spline matrices	source
utility.f	Additional math matrix in-core solvers	source
zaerom.f	Steady and unsteady aerodynamics processing for all of ZAERO's aerodynamic methods	source
zaerolib.o	ZONA's aerodynamic kernels	object

Note that all source code of ZAERO developed and integrated into ASTROS under this contract is being furnished to AFRL. The zaerolib.o code was developed prior to the current STTR Phases I & II and is ZONA Technology proprietary. This file is delivered in object code format only for specified computer platforms. To acquire updated object code for different computer platforms, please contact ZONA Technology at (602) 945-9988, POC: Darius Sarhaddi.

4.0 ZAERO ENGINEERING APPLICATION MODULES

Nine new engineering application modules have been developed as the ZAERO interface to ASTROS. The modules along with a brief functional descriptions are presented in Table 2.

Table 2. ZAERO Engineering Application Modules.

Module Name	Function
AEROGM	Aerodynamic Geometry Module
CONMOD	Control Surface Modes Generation
FLUTQHHZ	Process matrix [AJK] with normal modes for flutter
FLUTSENZ	To compute the sensitivities of active flutter constraints in the current boundary condition
FLUTTRAZ	Perform flutter analysis in the current boundary condition and to evaluate any flutter constraints if it is an optimization boundary condition with applied flutter constraints
QHHLGENZ	Compute the unsteady aerodynamic matricies in the modal dynamic degrees of freedom for gust analysis
SPLINZ	Generate the spline matrix that relates displacements and forces between the structural model and aerodynamic models
SZAERO	Generate steady aerodynamic AIC matrices and aerodynamic forces of unit configurations
UZAERO	Unsteady aeroelastic analysis preface

For ease of understanding, these new engineering modules are documented in the same format as those presented in the ASTROS Programmer's Manual (Ref 2). The modules presented provide the programmer a general description of the algorithm and clearly defines the module's arguments. In addition, the purpose, MAPOL calling sequence, FORTRAN subroutine name and method (i.e. function) of the module is presented. In cases of similar methods employed by modules to those of ASTROS (Version 11.0), the user is referred to the ASTROS Programmer's Manual (Ref 2).

Engineering Application Module: **AEROGM**

Entry Point: **AEROGM**

Purpose:

ZAERO geometry preface module.

MAPOL Calling Sequence:

```
CALL AEROGM  (  AECOMPZ, GEOMZA, AGRIDZ  );
```

AECOMPZ	A relation describing aerodynamic components (Output)
GEOMZA	A relation describing the aerodynamic boxes (Output)
AGRIDZ	A relation describing the corner points of aerodynamic boxes (Output)

Application Calling Sequence:

None

Method:

The **AEROGM** module processes all **BODY7** and **CAERO7** bulk data entries and computes the geometric data stored in the relational entites **AECOMPZ**, **GEOMZA**, and **AGRIDZ**. These relational entites are to be used by the **CONMOD**, **SPLINZ**, **UZAERO**, and **SZAERO** modules.

Design Requirements:

The **AEROGM** module is excuted in the preface phase. It is the aerodynamic geomety module for the **ZAERO** module.

Error Conditions:

None

Engineering Application Module: **CONMOD**

Entry Point: **CONMOD**

Purpose:

Control surface modes generation.

MAPOL Calling Sequence:

```
CALL CONMOD ( AECOMPZ, GEOMZA, [SCNTLG], [SCNTLK], [ACNTLG], [ACNTLK],
              [LMODG], [LMODK] );
```

AECOMPZ	A relation created by the **AEROGM** module describing aerodynamic components (Character, Input)
GEOMZA	A relation created by the **AEROGM** module describing the aerodynamic boxes (Character, Input)
[SCNTLG]	Matrix whose rows contain the symmetric control surface modes defined at the G-set D.O.F. and columns are associated with the **AESURFZ** bulk data entries. [SCNTLG] is used to compute the inertia loads by unit deflection angle of control surfaces. (Output)
[SCNTLK]	Matrix whose rows contain the symmetric control surface modes defined at the K-set D.O.F. and columns are associated with the **AESURFZ** bulk data entries. [SCNTLK] is used to compute the unsteady aerodynamic forces [**AJC**] and steady aerodynamic forces [**AIRFRC**] by unit deflection angle of the control surfaces. (Output)
[ACNTLG]	Same as [SCNTLG] but for antisymmetric control surface modes (Output)
[ACNTLK]	Same as [SCNTLK] but for antisymmetric control surface modes (Output)
[LMODEG]	Matrix whose rows contain the load modes at the G-set D.O.F. and columns are associated with the **LOADMOD** bulk data entries (Output)
[LMODEK]	Matrix whose rows contain the load modes at the K-set D.O.F. and columns are associated with the **LOADMOD** bulk data entries (Output)

Application Calling Sequence:

None

Method:

First, the **CONMOD** module processes all **AESURFZ** bulk data entries (if there are any) and generates the control surface modes due to unit deflection angle of the control surfaces about the hinge lines in both G-set and K-set D.O.F. If TYPE = 'SYM' or 'ASYM', the control surface modes are stored in [SCNTLG] and [SCNTLK]. If TYPE = 'ANTISYM', the control surface modes are stored in [**ACNTLG**] and [**ACNTLK**].

Next, the **CONMOD** module processes all **LOADMOD** bulk data entries (if there are any) and generates the load modes of each **LOADMOD**. The load modes are defined in the G-set and K-set D.O.F. and stored in each row of the matrix [**LMODEG**] and [**LMODEK**], respectively.

Design Requirements:

None

Error Conditions:

None

Engineering Application Module: **FLUTQHHZ**

Entry Point: **FLTQHZ**

Purpose:

Processes matrix **[AJK]** with normal modes for flutter.

MAPOL Calling Sequence:

```
CALL FLUTQHHZ (  NITER, BCID, SUB, ESIZE(BC), PSIZE(BC), [AJK], [SKJ],
                 [UGTKA], [PHIA], USET(BC), [TMN(BC)], [GSUBO(BC)], NGDR,
                 AECOMPZ, GEOMZA, [PHIKH], [QHHLFL(BC, SUB)], OAGRDDSP  );
```

NITER	Design iteration number (Integer, Input)
BCID	Boundary condition number (Integer, Input)
SUB	Flutter subcase number (Integer, Input)
ESIZE(BC)	Number of extra points for the current boundary condition (Integer, Input)
PSIZE(BC)	Number of physical degrees of freedom in the current boundary conditions (GSIZE+ESIZE) (Integer, Input)
[AJK]	Unsteady AIC matrices generated by the **UZAERO** module (Input)
[SKJ]	Integration matrix generated by the **UZAERO** module (Input)
[UGTKA]	The matrix of splining coefficients relating the aerodynamic pressures and forces at the structural grids and relating the structural displacements to the streamwise slopes of the aerodynamic boxes. **[UGTKA]** is reduced to the a-set DOF from **[UGTKG]**. (Input)
[PHIA]	Matrix of normal modes eigenvectors in the a-set (Input)
USET(BC)	Current boundary condition's unstructured entity of set definition masks (expanded to include extra points and any **GDR** scalar points) (Input)
[TMN(BC)]	Multipoint constraint transformation matrix for the current boundary condition (Input)
[GSUBO(BC)]	Static condensation or **GDR** reduction matrix for the current boundary condition (Input)
NGDR	Denotes dynamic reduction in the boundary condition = 0 No **GDR** = -1 **GDR** is used (Input, Integer)
AECOMPZ	A relation describing aerodynamic components created by the **AEROGM** module (Character, Input)

GEOMZA	A relation describing the aerodynamic boxes created by the **AEROGM** module (Character, Input)
[PHIKH]	A modal tranformation matrix that relates the box-on-box aerodynamic motions to unit displacements of the generalized structural coordinates (modes) (Output)
[QHHLFL(BC, SUB)]	A matrix containing the list of h x h unsteady aerodynamics matrices for the current flutter subcase related to the generalized (modal) coordinates and including control effectiveness (**CONEFFS**), extra points and **CONTROL** matrix inputs, where **BC** represents the MAPOL boundary condition loop index number (Output)
OAGRDDSP	A relation containing the structural eigenvectors (generalized DOF) mapped to the aerodynamic boxes for those **AIRDISP** requests in the Solution Control. These terms are the columns of **PHIKH** put in relational form to satisfy the output requests. (Output)

Application Calling Sequence:

None

Method:

FLUTQHHZ is very similar to the **FLUTQHHL** module (see **FLUTQHHL** Engineering Application Module of ASTROS Programmer's Manual for description of Method). There are only two differences between these two modules.

1. **FLUTQHHZ** reads in **[AJK]** and **[SJK]** matrices and computes the **QKK** matrices as

   $$[QKK] = [SJK]^T [AJK]^T$$

   then computes the generalized aerodynamic forces as

   $$[QHHLFL] = [PHIKH]^T [QKK] [PHIKH]$$

   therefore, the **[QKK]** matrix is a intermidiate matrix created in **FLUTQHHZ**. However, the actual procedure to compute **[QHHLFL]** in the **FLUTQHHZ** is described in ENTITY DESCRIPTIONS of **AJK**

2. **FLUTQHHZ** uses the relational entity **REUNMK** to retrieve the AIC matrices of the Mach number and associated reduced frequencies as defined in the **IDMK** of the **FLUTTER** bulk data entry.

Engineering Application Module: **FLUTSENZ**

Entry Point: **FLTSTZ**

Purpose:

To compute the sensitivities of active flutter constraints in the current boundary condition.

MAPOL Calling Sequence:

```
CALL FLUTSENZ (  NITER, BC, SUB, LOOP, GSIZEB, NDV, GLBDES, CONST, GMKCT,
                 DKVI, GMMCT, DMVI, CLAMBDA, LAMBDA, [QHHLFL(BC, SUB)],
                 [BHHFL(BC, SUB)], [KHHFL(BC, SUB)], [PHIG(BC)], [AMAT],
                 AEROZ  );
```

NITER	Design iteration number (Integer, Input)
BC	Boundary condition identification number (Integer, Input)
SUB	Flutter subcase number (Integer, Input)
LOOP	Logical flag indicating whether more flutter subcases exist in the current boundary condition (Logical, Input)
GSIZEB	The size of the structural set (Integer, Input)
NDV	The number of global design variables (Integer, Input)
GLBDES	Relation of global design vaiables (Character, Input)
CONST	Relation of constraint values (Character, Input)
GMKTC	Relation containing the connectivity data for the **DKVI** sensitivty matrix (Character, Input)
DKVI	Unstructured entity containing the stiffness design sensitivity matrix in a highly compressed format (Character, Input)
GMMCT	Relation containing connectivity data for **DMVI** sensitivity matrix (Character, Input)
DMVI	Unstructured entity containing the mass design sensitivity matrix in a highly compressed format (Character, Input)
CLAMBDA	Relation containing results of flutter analysis (Character, Input)
LAMBDA	Relation containing the output from the real eigenanalysis (Character, Input)
[QHHLFL(BC, SUB)]	A matrix containing the list of h x h unsteady aerodynamics matrices for the current flutter subcase related to the generalized (modal) coordinates and including control effectiveness (**CONEFFS**), extra points and **CONTROL** matrix inputs, where **BC** represents the MAPOL boundary condition loop index number (Input)

[MHHFL(BC, SUB)]	Modal mass matrix (Input)
[BHHFL(BC, SUB)]	Modal flutter damping matrix (Input)
[KHHFL(BC, SUB)]	Modal flutter stiffness matrix (Input)
[PHIG(BC)]	Matrix of real eigenvectors in the structural set (Input)
[AMAT]	Matrix of constraint sensitivities (Output)
AEROZ	Relation containing the definition of the aerodynamic coordinate system (Input)

Application Calling Sequence:

None

Method:

FLUTSENZ is very similar to the **FLUTSENS** module (see **FLUTSENZ** Engineering Application Module for description of Method). There is only one difference between these two modules. **FLUTSENZ** uses the relational entity **REUNMK** to retrieve the AIC matrices of the Mach number and associated reduced frequencies as defined in the **IDMK** of the **FLUTTER** bulk data entry.

Design Requirements:

The module assumes that at least one flutter subcase exists in the current boundary condition.

Error Conditions:

None.

Engineering Application Module: **FLUTTRAZ**

Entry Point: **FLTTAZ**

Purpose:

To perform flutter analyses in the current boundary condition and to evaluate any flutter constraints if the current boundary condition is an optimization boundary condition with applied flutter constraints.

MAPOL Calling Sequence:

```
CALL FLUTTRAZ  ( NITER, BCID, SUB, [QHHLFL(BC, SUB)], LAMBDA, HSIZE(BC),
                 ESIZE(BC), GMKCT, [MHHFL(BC, SUB)], [BHHFL(BC, SUB)],
                 KHHFL(BC, SUB)], CLAMBDA, AEROZ  );
```

`NITER`	Design iteration number (Integer, Input)
`BCID`	User defined boundary condition identification number (Integer, Input)
`SUB`	Flutter subcase number (ranging from 1 to the total number of **FLUTTER** subcases) of the subcase to be processed in this pass (Integer, Input)
`[QHHLFL(BC, SUB)]`	Matrix list of modal unsteady aerodynamic coefficients (Input)
`LAMBDA`	Relational entity containing the output from the real eigenanalysis (Character, Input)
`HSIZE(BC)`	Number of modal dynamic degrees of freedom in the current boundary condition (Input)
`ESIZE(BC)`	The number of extra point degrees of freedom in the current boundary condition (Integer, Input)
`[MHHFL(BC, SUB)]`	Modal mass matrix (Input)
`[BHHFL(BC, SUB)]`	Modal flutter damping matrix (Input)
`[KHHFL(BC, SUB)]`	Modal flutter stiffness matrix (Input)
`CLAMBDA`	Relation containing results of flutter analyses (Character, Input)
`AEROZ`	Relational entity of the configuration parameters defined by the **AEROZ** bulk data entry (Character, Input)

Application Calling Sequence:

None

Method:

FLUTTRAZ is very similar to the **FLUTTRAN** module (see **FLUTTRAN** Engineering Application Module of the ASTROS Programmer's Manual for a description of the Method). The difference is that rather than processing the **UNMK** unstructured entity, **FLUTTRAZ** reads the relational entity **REUNMK** for retrieving the Mach number and reduced frequency pairs.

Design Requirements:

The module assumes that at least one flutter subcase exists in the current boundary condition.

Error Conditions:

Referenced data on **FLUTTER** entries that do not exist on the database are flagged and the execution is terminated.

Engineering Application Module: **QHHLGENZ**

Entry Point: **QHJGEN**

Purpose:

To compute the unsteady aerodynamic matrices in the modal dynamic degrees of freedom for gust analysis.

MAPOL Calling Sequence:

```
CALL QHHLGENZ (  BC, ESIZE(BC), [AJK], [SKJ], [QGK], [UGTKA], [PHIA],
                 [PHIKH], [QHHL], [QHJL], AEROZ  );
```

BC	Boundary condition identification number (Integer, Input)
ESIZE(BC)	The number of extra point degrees of freedom in the boundary condition (Integer, Input)
[AJK]	Unsteady AIC matrices generated by the **UZAERO** module (Input)
[SKJ]	Integration matrix generated by the **UZAERO** module (Input)
[QGK]	A matrix contaning the intermediated gust vectors generated by the **UZAERO** module (Input)
[UGTKA]	The matrix of splining coefficients relating the aerodynamic pressures and forces at the structural grids and relating the structural displacements to the streamwise slopes of the aerodynamic boxes reduced to the a-set DOF. Generated by the **SPLINZ** module. (Input)
[PHIA]	Matrix of normal modes eigenvectors in the a-set (Input)
[PHIKH]	A modal tranformation matrix that relates the box-on-box aerodynamic motions to unit displacements of the generalized structural coordinates (modes) (Output)
[QHHL]	A matrix containing the list of h x h unsteady aerodynamics matrices of each reduced frequency for the current gust subcase related to the generalized (modal) coordinates (Output)
[QHJL]	A matrix containing the list of h x 1 unsteady hormonic gust vector of each reduced frequency (Output)
AEROZ	A relation containing the definition of the aerodynamic coordinate system (Input)

Application Calling Sequence:

None

Method:

QHHLGENZ is very similar to the **QHHLGEN** module (see **QHHLGEN** Engineering Application Module of the ASTROS Programmer's Manual for a description of the Method). There are only two differences between these two modules.

1. **QHHLGENZ** reads in **[AJK]** and **[SJK]** matrices and computes the **QKK** matrices as

   $$[QKK] = [SJK]^T [AJK]^T$$

   then computes the generalized aerodynamic forces as

   $$[QHHL] = [PHIKH]^T [QKK] [PHIKH]$$

   therefore, the **[QKK]** matrix is a intermediate matrix created in **QHHGENZ**.

2. The gust vector is computed as:

   $$[QHJL] = [PHIKH]^T [QGK] \exp(i*k/(\mathbf{REFC}/2.)*x_o)$$

   where k is the reducred frequency.
   REFC is the reference chord.

   and x_o is the location of thereference plane defined in the **GUST** bulk entry.

3. **QHHLGENZ** uses the relational entity **REUNMK** to retrieve the AIC matrices of the Mach number and associated reduced frequencies as defined in the **IDMK** of the **GUST** bulk data entry.

Engineering Application Module: **SPLINZ**

Entry Point: **SPLINZ**

Purpose:

Generates the spline matrix that relates displacements and forces between the structural model and the ZAERO aerodynamic model.

MAPOL Calling Sequence:

```
CALL SPLINZ ( GSIZEB, GEOMZA, AECOMPZ, AEROZ, [UGTKG]  );
```

GSIZEB	The number of degrees of freedom in the set of all structural GRID and SCALAR points (Integer, Input)
GEOMZA	A relation describing the aerodynamic boxes for the ZAERO model. The location of the box centroid, normal and pitch moment axis are given. It is used in splining the aerodynamics to the structure and to map responses back to the aerodynamic boxes. (Character, Input)
AECOMPZ	A relation describing aerodynamic components for the ZAERO model. It is used in splining the aerodynamics to the structural model. (Character, Input)
AEROZ	A relation created by the **AEROZ** bulk entry (Character, Input)
[UGTKG]	Spline matrix relating the structural displacements at G-set d.o.f to the displacements ans slopes at the K-set d.o.f of the aerodynamic boxes. (Output)

Application Calling Sequence:

None

Method:

The **SPLINZ** module is very similar to the **SPLINES** and **SPLINEU** modules (see ASTROS Programmer's Manual), except:

1. It only relates the aerodynamic boxes associated with **BODY7** and **CAERO7** to the structural model.
2. In addition to the **SPLINE1, SPLINE2** and **ATTACH** bulk data entries, it also reads the **SPLINE3** bulk data entry for 3D spline.
3. The spline matrix is used for both the steady and unsteady aeroelastic modules.

The spline matrix **[UGTKG]** is used for both steady aeroelastic analysis and dynamic aeroelastic analysis. For the definition of K-set d.o.f., please see entity descriptions of entity **UGTKG**.

Design Requirements:

None

Error Conditions:

1. Each aerodynamic box may appear on only one **SPLINE1, SPLINE2, SPLINE3** or **ATTACH** entry, although not all boxes need appear. Missing boxes will not influence the aeroelastic response.

2. Missing structural grids or aerodynamic elements appearing on the spline definitions will be flagged.

Engineering Application Module: **SZAERO**

Entry Point: **SZAERO**

Purpose:

Generates steady aerodynamic AIC matrices and aerodynamic forces of unit configuration parameters by the ZAERO module.

MAPOL Calling Sequence:

```
CALL SZAERO ( [AJK], MINDEX, LOOP, AECOMPZ, GEOMZA, AGRIDZ, STABCF,
[AICMAT(MINDEX)], [AAICMAT(MINDEX)], [AIRFRC(MINDEX)],
[SCNTLK], [ACNTLK] );
```

AJK	Unsteady AIC matrices generated by the **UZAERO** module (Input)
MINDEX	Mach number index for the current pass. Controls which Mach number/symmetry conditions will be processed in this pass by **SZAERO**. One pass for each unique Mach number will be performed with **MINDEX** incrementing by one until **SZAERO** returns **LOOP** = .FALSE. (Input)
LOOP	A logical flag set by **SZAERO** to indicate whether additional **MINDEX** subscripts are needed to complete the processing of all Mach number/symmetry conditions on all the **TRIM** bulk data entries. One pass for each unique Mach number will be performed with **MINDEX** incrementing by one until **SZAERO** returns **LOOP** = .FALSE. (Output)
AECOMPZ	A relation created by the **AEROGM** module describing aerodynamic components (Character, Input)
GEOMZA	A relation created by the **AEROGM** module describing the aerodynamic boxes (Character, Input)
AGRIDZ	A relation created by the **AEROGM** module describing the corner points of aerodynamic boxes (Character, Input)
STABCF	A relation of rigid aerodynamic stability coefficients for unit configuration parameters. The coefficients are stored in **STABCF** and the corresponding distributed forces are stored in **[AIRFRC(MINDEX)]**. The **STABCF** relation is used to pick the appropriate rigid loads from **[AIRFRC(MINDEX)]** when performing the aeroelastic trim as well as for retrieving the RIGID/FLEXIBLE stability coefficients for each configuration parameters. (Output)
[AICMAT(MINDEX)]	Matrix containing the steady aerodynamic influence coefficients for symmetric flight condition (Output)
[AAICMAT(MINDEX)]	Same as **[AICMAT(MINDEX)]** but for antisymmetric flight condition (Output)
[AIRFRC(MINDEX)]	Matrix containing the steady aerodynamic distributed forces for unit configuration parameters for the current Mach number index. If both symmetric and antisymmetric conditions exist for the Mach number, both sets of configuration parameters will coexist in **[AIRFRC]**. (Output)

[SCNTLK] Matrix (created by the **CONMOD** module) whose rows contain the symmetric control surface modes defined at the K-set D.O.F. and columns are associated with the **AESURFZ** bulk data entries. **[SCNTLK]** is used to compute the aerodynamic stabilitiy coefficients and distributed forces contained in **STABCF** and **[AIRFRC]** by unit deflection of control surfaces. (Input)

[ACNTLK] Same as **[SCNTLK]** but for antisymmetric control surface modes.

Application Calling Sequence:

None

Method:

The **SZAERO** module is very similar to the **STEADY** module (see ASTROS Programmer's Manual) except that **SZAERO** processes the aerodymanic geometry generated by the **AEROGM** module and computes the AIC matrices from ZONA6, ZONA7, ZTAIC, and ZONA7U methods for wing-body configurations. The output data format of **SZAERO** is identical to that of the **STEADY** module so that the output data can be directly used by the downstream steady aeroelatic trim modules.

The steady AIC matrices are obtained by taking the real part of the lowest reduced frequency of the matrix **[AJK]**, where **[AJK]** is generated by **UZAERO** module.

Design Requirements:

See **STEADY** module.

Error Conditions:

See **STEADY** module.

Engineering Application Module: **UZAERO**

Entry Point: **UZAERO**

Purpose:

Unsteady aeroelastic analysis preface by ZAERO module.

MAPOL Calling Sequence:

```
CALL UZAERO    (  AECOMPZ, GEOMZA, AGRIDZ, [AJK], [AJC], [AJL] [QGK], [SKJ],
                  [SCNTLK], [ACNTLK], [LMODEK]  );
```

AECOMPZ	A relation created by the **AEROGM** module describing aerodynamic components (Character, Input)
GEOMZA	A relation created by the **AEROGM** module describing the aerodynamic boxes (Character, Input)
AGRIDZ	A relation created by the **AEROGM** module describing the corner points of aerodynamic boxes (Character, Input)
[AJK]	Matrix containing the transposed unsteady aerodynamic influence coefficient (AIC) matrices for all Mach, and reduced frequency pairs defined in all **MKAEROZ** bulk data entries (Output)
[AJC]	Matrix containing the unsteady pressure in J-set D.O.F. on aerodynamic boxes due to the control surface modes for all Mach number and reduced frequency pairs defined in all **MKAEROZ** bulk data entries (Output)
[AJL]	Matrix containing the unsteady pressure in J-set D.O.F. on aerodynamic boxes due to the load modes for all Mach number and reduced frequency pairs defined in all **MKAEROZ** bulk data entries (Output)
[QGK]	Gust matrix containing the intermediated gust force vectors at the K-set D.O.F. for all Mach number and reduced frequency pairs defined in all **MKAEROZ** bulk data entries (Output)
[SKJ]	Integration matrix to take pressures in J-set D.O.F. to forces in K-set D.O.F (Output)
[SCNTLK]	Matrix (created by the **CONMOD** module) whose rows contain the symmetric control surface modes defined at the K-set D.O.F. and columns are associated with the **AESURFZ** bulk data entries. **[SCNTLK]** is used to compute the unsteady aerodynamic forces **[AJC]** by unit deflection of control surfaces. (Input)
[ACNTLK]	Same as **[SCNTLK]** but for antisymmetric control surface modes (Input)
[LMODEK]	Matrix (created by **CONMOD** module) whose rows contain load modes defined at the K-set D.O.F. and columns are associated with the **LOADMOD** bulk data entries. **[LMODEK]** is used to compute the unsteady aerodynamic forces **[AJL]** of the load modes. (Input)

Application Calling Sequence:

None

Method:

The **UZAERO** module first reads in the relational entity **AEROZ** to check the symmetric condition of the aerodynamic geomertry. If **XZSYM** = 'YES', the symmetric AIC and antisymmetric AIC matrices will be generated regardless of whether they are required for the downstream unsteady aeroelastic modules. The AIC matrices are generated according to the input sequence of **MKAEROZ** bulk data entries. Each **MKAEROZ** will produce a set of AIC matrices at the given Mach number and its associated list of reduced frquencies. The geometric data of the aerodynamic model is based on the relations **AECOMPZ**, **GEOMZA**, and **AGRIDZ**.

The AIC matrices of Mach, reduced frequency, symmetry pairs are stored in **[AJK]**. **[AJC]** is computed by:

$$[AJC] = [AJK]^T [\,[SCNTLK], [ACNTLK]\,]$$

pre-mutiplied **[AJC]** by $[SKJ]^T$ will yield the control surface aerodynamic forces at K-set D.O.F.

The intermediated gust force vector **[QGK]** is computed by:

$$[QGK]=[SKJ]^T\,[AJK]^T\,\{\exp(-i*K*X/(REFC/2.)\}$$

where K is the reduced frequency.
X is the aerodynamic box control point locations.
REFC is the reference chord.

[AJL] is computed by:

$$[AJL] = [AJK]^T[LMODEK]$$

pre-mutiplied **[AJL]** by $[SKJ]^T$ will yield the load mode aerodynamic forces at K-set D.O.F.

The method to retrieve the **[AJK]** and **[AJC]**, and **[AJL]** matrices of a given Mach number, reduced frquency, and symmetry pair is described in relational entity **REUNMK**.

Design Requirements:

Unlike the **AMP** module, the **UZAERO** module does not generate the **[QKK]** matrix.
The **[QKK]** matrix is computed by the **FLUTQHHZ** module from:

$$[QKK] = [SKJ]^T[AJK]^T$$

The unsteady forces due to control surface modes (defined as **[QKC]**) can be computed by:

$$[QKC] = [SKJ]^T[AJC]$$

Error Conditions:

None

5.0 ZAERO DATABASE ENTITY DESCRIPTIONS

To facilitate the communication of data among the ZAERO engineering application modules, fifteen new database entities (11 Matrix and 4 Relational) are created and are presented in Table 3.

Table 3. ZAERO Database Entities.

Entity Name	Description	Type
AJC	Basic name of the unsteady aerodynamic matrix containing unsteady pressure coefficients at J-set d.o.f. due to unit control surface deflections.	Matrix
QGK	Basic name of the unsteady aerodynamic gust force vector containing the intermediated unsteady forces at K-set d.o.f	Matrix
SKJ	Integration matrix relating the unsteady aerodynamic pressure coefficients at the J-set d.o.f. to the unsteady aerodynamic forces at the K-set d.o.f.	Matrix
AJK	Basic name of the unsteady aerodynamic AIC matrix relating the displacements at the K-set d.o.f to the pressure coefficients at the J-set d.o.f.	Matrix
ACNTLK	Displacements and slopes defined at K-set d.o.f. due to unit anti-symmetric control surface deflection.	Matrix
SCNTLK	Translational and rotational displacements defined at G-set d.o.f. due to unit symmetric control surface deflection.	Matrix
SCNTLG	Displacements and slopes defined at K-set d.o.f. due to unit symmetric control surface deflection.	Matrix
ACNTLG	Translational and rotational displacements defined at G-set d.o.f. due to unit anti-symmetric control surface deflection.	Matrix
LMODEG	Translational and rotational displacements defined at G-set d.o.f due to the load modes specified in bulk entries **LOADMOD**.	Matrix
LMODEK	Displacements and slopes defined at K-set d.o.f due to the load modes specified in bulk entries **LOADMOD**.	Matrix
UGTKG	Spline matrix relating the structural displacements at G-set d.o.f to the displacements and slopes at the K-set d.o.f of the aerodynamic boxes, but stored in the transposed form.	Matrix
AECOMPZ	Contains data on the aerodynamic components in the **CAERO7** and **BODY7** bulk entries.	Relation
GEOMZA	Contains data on the aerodynamic boxes of the **CAERO7** and **BODY7** bulk entries.	Relation
AGRIDZ	Contains data of the corner grid points on the **CAERO7** and **BODY7** boxes.	Relation
REUNMK	Contains the relations between the unsteady aerodynamic matrices generated by the **UZAERO** module to the bulk entries **MKAEROZ**.	Relation

The ZAERO database entities are documented similar to those in the ASTROS Programmer's Manual (Ref 2). A Usage section has been added to aide and clearly define to the programmer data stored on each database entitiy.

Entity: **AJC**

Entity Type: MATRIX

Description: Basic name of the unsteady aerodynamic matrix containing unsteady pressure coefficients at J-set d.o.f. due to unit control surface deflections. **AJC** is used during the aeroservoelastic analysis.

Matrix Form: Complex matrix with number of columns being equal to the number of control surfaces and J-set number of rows being equal to the number of J-set d.o.f.

Created by: **UZAERO**

Usage:

AJC contains a three characters string 'AJC' defined by **MAPOL**. To retrieve the **AJC** of a given Mach number, reduced frequency pair and symmetry condition, please see entity **REUNMK**.

The actual matrix name stored on the data base is $\mathbf{AJC}_{siijj}$,
where s='S' for symmetric or asymmetric case, ='A' for antisymmetric case.
ii=index of Mach number.
jj=index of reduced frequency.

The matrix **QKC** defined as the unsteady aerodynamic forces due to unit control surface deflections at K-set is computed by:

$$[\mathbf{QKC}]=[\mathbf{SKJ}]^T[\mathbf{AJC}_{siijj}]$$

The unsteady generalized aerodynamic control forces **[QHCLFL]** is computed by:

$$[\mathbf{QHCLFL}]=[\mathbf{PHIKH}]^T[\mathbf{QKC}]$$

where **[PHIKH]** is the modal matrix at K-set d.o.f.

Therefore the number of rows of **[QHCLFL]** is the number of modes. Each column of **[QHCLFL]** corresponds to the generalized aerodynamic control forces due to each of the bulk entry **AESURFZ** with TYPE=SYM for $\mathbf{AJC}_{siijj}$ and TYPE=ANTISYM for $\mathbf{AIC}_{aiijj}$.

Entity: **QGK**

Entity Type: MATRIX

Description: Basic name of the unsteady aerodynamic gust force vector containing the intermediated unsteady forces at K-set d.o.f. **QGK** is used by the aeroservoelastic gust analysis.

Matrix Form: Complex matrix with one column and K-set number of rows.

Created by: **UZAERO**

Usage:

QGK contains a three character string 'QGK' defined by **MAPOL**. To retrieve the **QGK** of a given Mach number, reduced frequency pair and symmetry condition, please see entity **REUNMK**.

The actual matrix name stored on the data base is $\mathbf{QGK}_{siijj}$,
where s='S' for symmetric or asymmetric case, ='A' for antisymmetric case.
ii=index of Mach number.
jj=index of reduced frequency.

The actual gust generalized forces in modal space is computed by:

$$[QGK_{siijj}] = [QGK_{siijj}] * \exp(i*k*x_o/(REFC/2.))$$

where x_o is the location of the reference plane defined in the bulk entry **GUST**.
k is the corresponding reduced frequency.
and REFC is the reference chord defined in bulk entry **AEROZ**.

Entity:	**SKJ**
Entity Type:	MATRIX
Description:	Integration matrix relating the unsteady aerodynamic pressure coefficients at the J-set d.o.f. to the unsteady aerodynamic forces at the K-set d.o.f.
Matrix Form:	Real matrix with J-set number of column and K-set number of rows but stored in the transposed form.
Created by:	**UZAERO**

Usage:

SKJ depends on the geometry of the aerodynamic model only and is independent of Mach number and reduced frequency.

Entity: **AJK**

Entity Type: MATRIX

Description: Basic name of the unsteady aerodynamic AIC matrix relating the displacements at the K-set d.o.f to the pressure coefficients at the J-set d.o.f.

Matrix Form: Complex matrix with K-set number of columns and J-set number of rows but stored in the transposed form.

Created by: **UZAERO**

Usage:

AJK contains a three characters string 'AJK' defined by **MAPOL**. To retrieve the **AJK** of a given Mach number, reduced frequency pair and symmetry condition, please see entity **REUNMK**.

The actual matrix name stored on the data base is $\mathbf{AJK}_{siijj}$,
where s='S' for symmetric or asymmetric case, ='A' for antisymmetric case.
ii=index of Mach number.
jj=ndex of reduced frequency.

The matrix **QKK** relating displacements at K-set to unsteady aerodynamic forces at K-set is computed by:

$$[QKK]=[SKJ]^{T}\,[AJK_{siijj}]^{T}$$

The unsteady generalized aerodynamic forces **[QHHLFL]** is computed by:

$$[QHHLFL]=[PHIKH]^{T}\,[QKK][PHIKH]$$

where **[PHIKH]** is the modal matrix at K-set d.o.f.

However, in the **FLUTQHHZ** module and **QHHLGENZ** module, **[QHHLFL]** is computed by the following procedure:

The unsteady aerodynamic pressure coefficients **[CP]** at J-set d.o.f. is first obtained

$$[CP]=[AJK_{siijj}]^{T}[PHIKH]$$

Then, the aerodynamic forces at K-set d.o.f are computed:

$$[FORCE]=[SKJ]^{T}[CP]$$

Finally, the generalized aerodynamic forces are computed:

$$[QHHLFL]=[PHIHK]^{T}[FORCE]$$

Matrices **[CP]** and **[FORCE]** are deleted after **[QHHLFL]** is obtained.

Entity:	**ACNTLK**
Entity Type:	MATRIX
Description:	Displacements and slopes defined at K-set d.o.f. due to unit anti-symmetric control surface deflection. Each column is corresponding to each **AESURFZ** bulk entry with TYPE=ANTISYM.
Matrix Form:	Real matrix with K-set number of rows and number of columns being equal to the number of **AESURFZ** bulk entries with TYPE=ANTISYM.
Created by:	**CONMOD**

Usage:

1. **ACNTLK** is used by both **UZAERO** and **SZAERO** modules.

   For the **UZAERO** module, it generates the **[AJC]** matrix for all **MKAEROZ** bulk entries by:

   $$[AJC]=[AJK]^T[ACNTLK]$$

   For the **SZAERO** module, it generates the matrix **[AIRFRC]** and the aerodynamic stability coefficients of control surfaces (stored in relation **STABCF**) for each **TRIM** bulk entry by:

   $$[AIRFRC]=[AAICMAT]^T[ACNTLK]$$

2. **ACNTLK** does not exist if there are no **AESURFZ** with TYPE=ANTISYM.

Entity:	**SCNTLK**
Entity Type:	MATRIX
Description:	Displacements and slopes defined at K-set d.o.f. due to unit symmetric control surface deflection. Each column is corresponds to each **AESURFZ** bulk entry with TYPE=SYM or ASYM.
Matrix Form:	Real matrix with K-set number of rows and number of columns being equal to the number of **AESURFZ** bulk entries with TYPE=SYM or ASYM.
Created by:	**CONMOD**

Usage:

1. **SCNTLK** is used by both the **UZAERO** and **SZAERO** modules.

   For **UZAERO** module, it generates the **[AJC]** matrix for all **MKAEROZ** bulk entries by:

   $$[AJC]=[AJK]^T[SCNTLK]$$

   For the **SZAERO** module, it generates the matrix **[AIRFRC]** and the aerodynamic stability coefficients of control surfaces (stored in relation **STABCF**) for each **TRIM** bulk entry by:

   $$[AIRFRC]=[AICMAT]^T[SCNTLK]$$

2. **SCNTLK** does not exist if there are no **AESURFZ** with TYPE=SYM or ASYM.

Entity:	**SCNTLG**
Entity Type:	MATRIX
Description:	Translational and rotational displacements defined at G-set d.o.f. due to unit symmetric control surface deflection. Each column corresponds to an **AESURFZ** bulk entry with TYPE=SYM or ASYM.
Matrix Form:	Real matrix with G-set number of rows and number of columns being equal to the number of **AESURFZ** bulk entries with TYPE=SYM or ASYM.
Created by:	**CONMOD**

Usage:

1. **SCNTLG** is used to compute the inertial matrix of the control surfaces in modal space by:

   $[PHIG]^T[MGG][SCNTLG]$ in G-set d.o.f.

   or

   $[PHIA]^T[MAA][SCNTLA]$ in A-set d.o.f. Where **[SCNTLA]** can be computed by the reduction of **[SCNTLG]** from G-set to A-set.

2. **SCNTLG** does not exist if there are no **AESURFZ** with TYPE=SYM or ASYM.

Entity:	**ACNTLG**
Entity Type:	MATRIX
Description:	Translational and rotational displacements defined at G-set d.o.f. due to unit anti-symmetric control surface deflection. Each column corresponds to an **AESURFZ** bulk entry with TYPE=ANTISYM.
Matrix Form:	Real matrix with G-set number of rows and number of columns being equal to the number of **AESURFZ** bulk entries with TYPE=ANTISYM.
Created by:	**CONMOD**

Usage:

1. **ACNTLG** is used to compute the inertial matrix of the control surfaces in modal space by:

   [PHIG]^T**[MGG][ACNTLG]** in G-set d.o.f.

   or

   [PHIA]^T**[MAA][ACNTLA]** in A-set d.o.f. Where **[ACNTLA]** can be computed by the reduction of **[ACNTLG]** from G-set to A-set.

2. **ACNTLG** does not exist if there are no **AESURFZ** with TYPE=ANTISYM.

Entity:	**LMODEG**
Entity Type:	MATRIX
Description:	Translational and rotational displacements defined at G-set d.o.f due to the load modes specified in bulk entries **LOADMOD**.
Matrix Form:	Real matrix with G-set number of rows and number of columns being equal to the number of **LOADMOD** bulk entries.
Created by:	**CONMOD**

Usage:

1. **LMODEG** is used to compute the sectional forces or moments at the structural grid points defined by the **LOADMOD** bulk entries. **LMODEG** can be reduced from G-set to A-set d.o.f. by the A-set reduction procedures.

2. **LMODEG** does not exist if there are no **LOADMOD** bulk data entries.

Entity:	**LMODEK**
Entity Type:	MATRIX
Description:	Displacements and slopes defined at K-set d.o.f due to the load modes specified in bulk entries **LOADMOD**.
Matrix Form:	Real matrix with K-set number of rows and number of columns being equal to the number of **LOADMOD** bulk entries.
Created by:	**CONMOD**

Usage:

1. **LMODEK** is used to compute the sectional forces or moments at the aerodynamic boxes defined by the **LOADMOD** bulk entries.

2. **LMODEK** does not exist if there are no **LOADMOD** bulk data entries.

Entity:	**UGTKG**
Entity Type:	MATRIX
Description:	Spline matrix relating the structural displacements at G-set d.o.f to the displacements and slopes at the K-set d.o.f of the aerodynamic boxes, but stored in the transposed form.
Matrix Form:	Real matrix with G-set number of rows and K-set number of columns.
Created by:	**SPLINZ**

Usage:

1. The definition of K-set d.o.f. is:

   For each aerodynamic box, six d.o.f.'s are assigned and defined as:

   {T1, T2, T3, d(T1)/dx, d(T2)/dx, d(T3)/dx}, where T1, T2, and T3 are the displacements at the centroid of the aerodynamic box along x, y, and z directions, respectively. d( )/dx denotes as the slope of ( ) with respect to the free stream direction (the x-axis of the aerodynamic coordinates).

   Therefore, for N number of aerodynamic boxes (number of J-set d.o.f.'s = N), number of K-set d.o.f.'s = 6 * N.

2. **[UGTKG]** can be reduced to **[UGTKA]** by the A-set reduction procedures, where **[UGTKA]** is used to transform the displacements at A-set to K-set and transform the aerodynamic forces from K-set to A-set by the transposed of **[UGTKA]**.

3. **[UGTKG]** is computed according to the **SPLINE1**, **SPLINE2**, **SPLINE3**, and **ATTACH** bulk entries.

Entity: **AECOMPZ**

Entity Type: Relation

Description: Contains data on the aerodynamic components in the **CAERO7** and **BODY7** bulk data entries.

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
ACID	Integer>0	Identification number of **CAERO7** or **BODY7** bulk entries.
MACROTYP	Text(8)	Either 'CAERO7' or 'BODY7'.
GROUP	Integer	Identification number of the **ACOORD** bulk entry.
ACMPNT	Text(8)	Component type. One of WING or BODY.
TYPE	Integer>0	TYPE=2 for **CAERO7**, 3 for **BODY7**.
FIINTID	Integer>0	First internal aerodynamic box identification number.
NCBOX	Integer>0	Number of chordwise boxes for **CAERO7**. =1 for **BODY7**.
NSBOX	Integer>0	Number of spanwise boxes for **CAERO7**. Number of boxes for **BODY7**.
BNDRY	R Vector(12)	For **CAERO7**: BNDRY(i), i=1,3: x, y, z of leading edge at root. BNDRY(i), i=4,6: x, y, z of trailing edge at root. BNDRY(i), i=7,9: x, y, z of leading edge at tip. BNDRY(i), i=10,12: x, y, z of trailing edge at tip. For **BODY7**: BNDRY(i), i=1,3: x, y, z of the nose. BNDRY(4): base pressure of the body wake. BNDRY(5): X location of the steady point singularity of the body wake. BNDRY(6): X location of the unsteady point singularity of the body wake. BNDRY(i), i=7,8: Y and Z offset for the point singularity of the body wake. BNDRY(9): Body length. BNDRY(10): Flag for body wake. (Integer) BNDRY(11): Number of inlet boxes. (Integer) BNDRY(12): Number of wake boxes on the body.
WCOS		For **CAERO7**: Cos(theta), where theta = dihedral angle. For **BODY7**: Number of segments. (Integer)
WSIN		For **CAERO7**: Sin(theta), where theta = dihedral angle. For **BODY7**: Not used.
IWING	Integer	Flag for vertical fin on the X-Z plane. =0: yes. =1, no.
ATTR	Integer	=0: **CAERO7** root is not attached to **BODY7**. >0: **CAERO7** root is attached to **BODY7** with ID=ATTR. Not used for **BODY7**.
YRB	Real	Y location of the center line of **BODY7** to which the **CAERO7** root is attached.
ZRB	Real	Z location of the center line of **BODY7** to which the **CAERO7** root is attached.

FLCOSR	Real	Cos(theta), where theta is the dihedral angle of the vortex-carry-through boxes at root.
FLSINR	Real	Sin(theta), where theta is the dihedral angle of the vortex-carry-through boxes at root.
ATTT	Integer	=0: **CAERO7** Tip is not attached to **BODY7**. >0: **CAERO7** Tip is attached to **BODY7** with ID=ATTT Not used for **BODY7**.
YTB	Real	Y location of the center line of **BODY7** if **CAERO7** tip is attached to it.
ZTB	Real	Z location of the center line of **BODY7** if **CAERO7** root is attached to it.
FLCOST	Real	Cos(theta), where theta is the dihedral angle of the vortex-carry-through boxes at tip.
FLSINT	Real	Sin(theta), where theta is the dihedral angle of the vortex-carry-through boxes at tip.
LABEL	Text(8)	Label of **CAERO7** or **BODY7** bulk entries.

Created by: **AEROGM**

Usage:

AECOMPZ is used by **SPLINZ**, **UZAERO** and **SZAERO** modules.

Entity: **GEOMZA**

Entity Type: Relation

Description: Contains data on the aerodynamic boxes of the **CAERO7** and **BODY7** bulk data entries.

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
MACROID	Integer	Component identification number of the associated **CAERO7** or **BODY7**.
ACMPNT	Text(8)	='FUSEL' for **BODY7** box, ='WING' for **CAERO7** box.
NDOF	Integer	=3 for **BODY7** box, =2 for **CAERO7** box.
EXTID	Integer	External identification number of the box.
INTID	Integer	Internal identification number of the box.
AREA	Real	Area of the box.
X	Real	X location of centroid of the box.
Y	Real	Y location of centroid of the box.
Z	Real	Z location of centroid of the box.
N1	Real	X component of the box normal in basic coordinates.
N2	Real	Y component of the box normal in basic coordinates.
N3	Real	Z component of the box normal in basic coordinates.
R1	Real	X component of the box local pitch axis in basic coordinates.
R2	Real	Y component of the box local pitch axis in basic coordinates.
R3	Real	Z component of the box local pitch axis in basic coordinates.
RTHETA	Real	For **BODY7** box: dihedral angel of the box. For **CAERO7** box: Thickness slope at 50% chord.
RDELTA	Real	For **BODY7** box: Inclination angel of the box. For **CAERO7** box: Camber slope at 50% chord.
CHORD	Real	Chord length.
ID1	Integer	Aerodynamic grid identification number at left hand side corner of the box leading edge.
ID2	Integer	Aerodynamic grid identification number at left hand side corner of the box trailing edge.
ID3	Integer	Aerodynamic grid identification number at right hand side corner of the box leading edge.
ID4	Integer	Aerodynamic grid identification number at right hand side corner of the box trailing edge.
CAM85	Real	Camber slope at 85% chord for **CAERO7** box. Not used for **BODY7** box.
CAM95	Real	Camber slope at 95% chord for **CAERO7** box. Not used for **BODY7** box.
DZX85	Real	Thickness slope at 85% chord for **CAERO7** box. Not used for **BODY7** box.
DZX95	Real	Thickness slope at 95% chord for **CAERO7** box. Not used for **BODY7** box.

DZXLE	Real	Thickness slope at leading edge of the mid-chord for **CAERO7** box. Not used for **BODY7** box.
DZXTE	Real	Thickness slope at trailing edge of the mid-chord for **CAERO7** box. Inlet panel flow ratio in percentage for **BODY7** box.
IWAKE	Integer	For **BODY7** box=1, box is inlet panel. =0, box is not inlet panel. Not used for **CAERO7** box.

Created by: **AEROGM**

Usage:

GEOMZA is used by **SPLINZ**, **UZAERO** and **SZAERO** modules.

Entity: **AGRIDZ**

Entity Type: Relation

Description: Contains data of the corner grid points on the **CAERO7** and **BODY7** boxes.

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
EXTID	Integer>0	External identification of the grid point.
INTID	Integer>0	Internal identification of the grid point.
CORD	Integer	Identification number of **ACOORD** bulk entry.
X	Real	X location of the grid point.
Y	Real	Y location of the grid point.
Z	Real	Z location of the grid point.

Created by: **AEROGM**

Usage:

AGRIDZ is used by **UZAERO** and **SZAERO** modules.

Entity: **REUNMK**

Entity Type: Relation

Description: Contains the relations between the unsteady aerodynamic matrices generated by the **UZAERO** module to the bulk entries **MKAEROZ**.

Relation Attributes:

NAME	TYPE/KEY	DESCRIPTION
IDMK	Integer>0	Identification number specified in the bulk entries **MKAEROZ**.
MACH	Real≥0.	Mach number specified in bulk entries **MKAEROZ**.
METHOD	Integer	Method flag specified in bulk entries **MKAEROZ**.
SYMXZ	Integer	Symmetry flag. SYMXZ=1 for symmetric case, =-1 for antisymmetric case, =0 for asymmetric case.
ALPHA	Real	Angle of attack specified in the **TRIMFLT** bulk entry of the current **MKAEROZ**.
BETA	Real	Side slip angle specified in the **TRIMFLT** bulk entry of the current **MKAEROZ**.
PRATE	Real	Non-dimensional roll rate specified in the **TRIMFLT** bulk entry of the current **MKAEROZ** bulk entry of the current **MKAEROZ**.
QRATE	Real	Nondimensional pitch rate specified in the **TRIMFLT** bulk entry of the current **MKAEROZ**.
RRATE	Real	A non-dimensional yaw rate specified in the **TRIMFLT** bulk entry of the current **MKAEROZ**.
MINDEX	Integer>0	Index of the **MKAEROZ** bulk entry ranging from 1 to the number of the **MKAEROZ** bulk entries.
KINDEX	Integer>0	Index of the reduced frequency ranging from 1 to the number of reduced frequencies specified in the current **MKAEROZ**.
RFREQ	Real>0.0	The KINDEX'th reduced frequency specified in the current **MKAEROZ**.

Created by: **UZAERO**

Usage:

The **UZAERO** module generates the unsteady aerodymanic matrices **[AJK]**, **[AJC]**, and **[QGK]** of all **MKAEROZ** bulk entries in the input file regardless of whether or not they are required for the downstream unsteady aeroelastic modules. To retrieve these matrices, please see the example on the following page:

For a given pair of IDMK and SYMXZ found in either the **FLUTTER** or **GUST** bulk entry, to retrieve the corresponing matrix [AJK]:

```
      CHARACTER*8 UNLIST(12),NAME
      DATA UNLIST/'IDMK','MACH','METHOD','SYMXZ','ALPHA','BETA','PRATE','QRATE'
     *                          ,'RRATE','MINDEX','KINDEX','RFREQ'/
      INTEGER INFO(20),IGET(12),MINDEX(100),KINDEX(100),SYMXZ
      REAL RGET(12),K(100),MACH
      EQUIVALENCE (RGET(1),IGET(1))
      CHARACTER*1 S
      CALL DBOPEN(REUNMK,INFO,'RO','NOFLUSH',ISTAT)
      CALL REPROJ(RENUMK,12,UNLIST)
      NMK=INFO(3)
C                NMK = total number of MKAEROZ bulk entries.
      INDEX=0
      DO I=1,NMK
              CALL REGET(REUNMK,IGET,ISTAT)
               IF(IDMK.EQ.IGET(1)) THEN
                       INDEX=INDEX+1
                       MACH=REGET(2)
                       METHOD=IGET(3)
                        ISYM=IGET(4)
                       MINDEX(INDEX)=IGET(10)
                       KINDEX(INDEX)=IGET(11)
                       K(INDEX)=RGET(12)
               ENDIF
      ENDDO
      CALL DBCLOS(REUNMK)
      KTOTAL=INDEX
C  KTOTAL is the total number of reduced frequencies specified in the MKAEROZ bulk entry
C  with IDMK as the identification number.
C  IF one wishes to retrive the [AJK] matrix of the second reduced frequency, do the
C  following:
      KTH=2
      IF(SYMXZ.EQ.1.OR.SYMXZ.EQ.0) THEN
              S='S'
      ELSE
              S='A'
      ENDIF
C Subroutine MYNAME is an utility routine to assemble the matrix name.
C INPUT:  AJK          A three characters string contains the basic name of the matrix.
C               S='S' for symmetric or asymmetric case, ='A' for antisymmetric case.
C              MINDEX(KTH)      KTH'th Mach number index  found in the REUNMK realtion.
C              KINDEX(KTH)     KTH'th reduced frequency index found in the REUNMK relation.
C OUTPUT: A character*8 string of the matrix created by UZAERO module with the form:
C                      AJKsiijj, where s=S, ii=MINDEX(KTH), and jj=KINDEX(KTH)
      CALL MYNAME(AJK,S,MINDEX(KTH),KINDEX(KTH),NAME)
C Now, NAME is the matrix name of the AIC matrix of the corresponding Mach number and
C reduced frequency.
       CALL MYNAME(AJC,S,MINDEX(KTH),KINDEX(KTH),NAME)
C Now, NAME is the matrix name of the control surface forces matrix of the corresponding
C Mach number and reduced frequency.
       CALL MYNAME(QGK,S,MINDEX(KTH),KINDEX(KTH),NAME)
C Now, NAME is the matrix name of the gust force matrix of the corresponding
C Mach number and reduced frequency.
C      .....................
C      .....................
```

6.0 REFERENCES

1. D.J. Neill, D.L. Herendeen, "ASTROS User's Manual," Volume I, WL-TR-96-3004, May 1995.
2. D.J. Neill, D.L. Herendeen, R.L. Hoesly, "ASTROS Programmer's Manual," Volume II, WL-TR-93-3038, March 1993.
3. Johnson, E.H. and Venkayya, V.B., "Automated Structural Optimization System (ASTROS), Theoretical Manual," AFWAL-TR-88-3028, Vol. 1, December 1988.

APPENDIX A

ZAERO FUNCTIONAL MODULE DEFINITION

(MODDEF.DAT)

The following is a list of all ZAERO module definitions added to ASTROS and found in file **MODDEF.DAT**.

```
AEROGM      3
102   7   7   7
C
C    AERO GEOMTRY FOR ZAERO MODULE
C NOTE: ALPHABETICAL ORDER IN FILE MODDEF.DAT IS NOT REQUIRED
C
      CALL AEROGM ( EP(1), EP(2), EP(3) )
END
```

```
CONMOD      8
 102   7   7   8   8   8   8   8   8
C
C    ZAERO CONTROL MODE GENERATOR
C
      CALL CONMOD ( EP(1), EP(2), EP(3), EP(4) ,EP(5),EP(6),EP(7),
     1              EP(8) )
END
```

```
FLUTQHHZ  18
 102  -1   1   1   1   1   8   8   8   8   9   8   8   1   7   7   8   8   7
C
C     PROCESS THE 'FLUTQHHL' MODULE - FLUTTER AEROMATRIX PROCESSOR
C
      CALL FLTQHZ ( IP(1), IP(2), IP(3), IP(4), IP(5), EP(6), EP(7),
     1              EP(8), EP(9), EP(10), EP(11), EP(12), IP(13),
     2              EP(14), EP(15), EP(16), EP(17) ,EP(18))
END
```

```
FLUTSENZ  21
 102   1   1   1   4   1   1   7   7   7   9   7   9   7   7   8   8   8   8   8
   8   7
C
C     PROCESS THE 'FLTSTY' MODULE TO OBTAIN FLUTTER CONST. SENSITIV.
C
      CALL FLTSTZ ( IP(1), IP(2), IP(3), LP(4), IP(5), IP(6), EP(7),
     *              EP(8), EP(9), EP(10), EP(11), EP(12), EP(13),
     *              EP(14), EP(15), EP(16), EP(17), EP(18), EP(19),
     *              EP(20), EP(21) )
END
```

```
FLUTTRAZ  13
 102  -1   1   1   8   7   1   1   8   8   8   7  -7   7
C
C     PROCESS THE 'FLUTAN' MODULE TO PERFORM FLUTTER ANALYSIS
C
      CALL FLUTAZ ( IP(1), IP(2), IP(3), EP(4), EP(5), IP(6), IP(7),
     1              EP(8), EP(9), EP(10), EP(11), EP(12),EP(13) )
END
```

```
QHHLGENZ  11
 102   1   1   8   8   8   8   8   8   8   8   7
C
C     'QHHLGENZ'  - GENERATE THE QHH MATRIX LIST FOR FLUTTER ANALYSIS
C
      CALL QHJGEN ( IP(1), IP(2), EP(3), EP(4), EP(5), EP(6),
     1              EP(7), EP(8), EP(9), EP(10), EP(11) )
END
```

```
SPLINZ      5
 102   1   7   7   7   8
C
C     PROCESS THE UNSTEADY AERODYNAMIC SPLINE
C
      CALL SPLINZ ( IP(1), EP(2), EP(3), EP(4), EP(5) )
END
```

```
SZAERO    12
 102   8   1   4   7   7   7   7   8   8   8   8   8
C
C     PROCESS ZAERO STEADY AERODYNAMICS
C       (PREFACE TO STATIC AEROELASTICITY DISCIPLINE)
C
      CALL SZAERO ( EP(1), IP(2), LP(3), EP(4), EP(5), EP(6), EP(7),
     1              EP(8), EP(9), EP(10),EP(11),EP(12) )
END
```

```
UZAERO    11
 102   7   7   7   8   8   8   8   8   8   8   8
C
C    AIC GENERATION BY ZAERO MODULE
C
      CALL UZAERO ( EP(1), EP(2), EP(3), EP(4), EP(5), EP(6),
     1              EP(7), EP(8), EP(9), EP(10),EP(11) )
END
```

```
INPUT4     6
 102  -1   1   7   8   8   8
C
C    READ MODAL RESULTS FROM NASTRAN OUTPUT4 SOLUTION
C    AND REPLACE THE ASTROS DATABASE MATRICIES KAA, MAA, PHIA
C    AND RELATION LAMBDA
C
      CALL INPUT4 (IP(1),IP(2),EP(3),EP(4),EP(5),EP(6))
END
```

APPENDIX B

ASTROS* MAPOL SEQUENCE LISTING

The following ASTROS* MAPOL sequence listing documents all changes made to the original ASTROS MAPOL sequence. All newly added lines and commented lines for integration of ZAERO into ASTROS are highlighted in boldfaced text. Arrows are also used at the ends of the lines to demarcate the beginning and ending of changes.

ASTROS* MAPOL Sequence Listing:

```
***** MAPOL SOURCE CODE LISTING *****
              STAT.LEVL
                1    1!$***$                                                                    !
                2    1!$ CSCIID <@(#) MC0083-MAPOLSEQ 11.1 4/29/94 17:00:35> $                  !
                3    1!$***$                                                                    !
                4    1!$*************************************************************************$!
                5    1!$                          EXECUTIVE SEQUENCE FOR ASTROS                  $!
                6    1!$*************************************************************************$!
                7    1!$*************************************************************************$!
                8    1!$            CONSTANTS FOR SDCOMP SET SINGULARITY MESSAGES                $!
                9    1!$*************************************************************************$!
               10    1!INTEGER    SINGOSET,    SINGASET,    SINGLSET;                           !
               11    1!$*************************************************************************$!
               12    1!$            VARIABLE DECLARATION SEGMENT                                 $!
               13    1!$*************************************************************************$!
               14    1!$                                                                         $!
               15    1!INTEGER    GSIZE,       NDV,         NITER,       BC,                    !
               16    1!           ESIZE(1000), PSIZE(1000), GSIZEB;                             !
               17    1!REAL       CTL,         CTLMIN;                                          !
               18    1!LOGICAL    GLBCNVRG,    APPCNVRG,    PFLAG;                           !
               19    1!UNSTRUCT   DCENT,       GRIDTEMP,    SMPLOD;                          !
               20    1!RELATION   DESHIST,     CONST,       MPPARM,      CONVERT,     OCPARM,  !
               21    1!           MFORM,       GRID,        SPOINT,      EPOINT,      SEQGP,   !
               22    1!           BGPDT(1000), CSTM,        FORCE,       FORCE1,      MOMENT,  !
               23    1!           MOMENT1,     PLOAD,       GRAV,        LOAD,        EIGR,    !
               24    1!           TEMP,        TEMPD,       OPNLBUCK,    OEULBUCK,             !
               25    1!           CORD1C,      CORD1R,      CORD1S,      CORD2C,      CORD2R,  !
               26    1!           CORD2S,      GPWGGRID,    OGPWG,       GRADIENT;              !
               27    1!$                                                                         $!
               28    1!$*************************************************************************$!
               29    1!$          DECLARATIONS FOR MODULE MKUSET                                 $!
               30    1!$*************************************************************************$!
               31    1!$                                                                         $!
               32    1!UNSTRUCT   USET(1000),  GPST(1000);                                      !
               33    1!RELATION   SPC,         SPC1,        SPCADD,      MPC,         MPCADD,  !
               34    1!           ASET,        ASET1,       OMIT,        OMIT1,       SUPORT,  !
               35    1!           JSET,        JSET1,       RBAR,  RBE1, RBE2,  RBE3, RROD;    !
               36    1!MATRIX     [PGMN(1000)],[PNSF(1000)],[PFOA(1000)],[PARL(1000)],[TMN(1000)],!
               37    1!           [YS(1000)];                                                   !
               38    1!MATRIX     [PGMNS(1000)],            [PNSFS(1000)],             [PFOAS(1000)],!
               39    1!           [PARLS(1000)];                                                !
               40    1!$                                                                         $!
               41    1!$*************************************************************************$!
               42    1!$          DECLARATIONS FOR MODULES MAKEST AND EMG                        $!
               43    1!$*************************************************************************$!
               44    1!$                                                                         $!
               45    1!UNSTRUCT   TREF,        DVSIZE,      PCOMPS;                          !
               46    1!IUNSTRUCT  KELM,        MELM,        TELM;                            !
               47    1!RELATION   CQDMEM1,     QDMM1EST,    CROD,        CONROD,      RODEST,  !
               48    1!           CSHEAR,      SHEAREST,    CTRMEM,      TRMEMEST,    CMASS1,  !
               49    1!           CMASS2,      MASSEST,     CONM1,       CONM1EST,    CONM2,   !
               50    1!           CONM2EST,    CBAR,        BEAMEST,     CQUAD4,      QUAD4EST,!
               51    1!           CIHEX1,      IHEX1EST,    CIHEX2,      IHEX2EST,    CIHEX3,  !
               52    1!           IHEX3EST,    CELAS1,      CELAS2,      ELASEST,              !
               53    1!           PCOMP,       PQDMEM1,     PROD,        PSHEAR,               !
               54    1!           PTRMEM,      PMASS,       PELAS,       PBAR,        PSHELL,  !
               55    1!           PCOMP1,      PCOMP2,      PIHEX,       MAT1,        MAT2,    !
               56    1!           MAT8,        MAT9,        CTRIA3,      TRIA3EST;             !
               57    1!$                                                                         $!
               58    1!$*************************************************************************$!
               59    1!$          DECLARATIONS FOR DESIGN VARIABLES/CONSTRAINTS AND LINKING       $!
               60    1!$*************************************************************************$!
               61    1!$                                                                         $!
               62    1!RELATION   DESELM,      DESVARP,     DESVARS,     PLIST,       ELIST,   !
               63    1!           SHAPE,       SHPGEN;                                          !
               64    1!RELATION   DCONVM,      DCONTW,      DCONEP,      DCONFT,      DCONVMM, !
```

```
1!            DCONTWM,     DCONEPM,     DCONFTM,     DCONVMP,     DCONTWP,
1!            DCONEPP,     DCONFTP,     DCONALE,     DCONCLA,     DCONFLT,
1!            DCONTRM,     DCONSCF;
1!RELATION    DCONDSP,     DCONFRQ,     DCONTHK,     DCONTH2;
1!RELATION    DCONPMN,     DCONLMN,     DCONLAM;
1!RELATION    DCONBK,      DCONBKE;
1!RELATION    GLBDES,      DESLINK,     TFIXED,      LOCLVAR,     DVCT;
1!MATRIX      [PTRANS];
1!IMATRIX     [PMINT],     [PMAXT],     [SMAT];
1!$
1!$*************************************************************************
1!$          DECLARATIONS FOR OUTPUT FILE PROCESSING  (EDR/OFP)
1!$*************************************************************************
1!$
1!RELATION    GRIDLIST,    MODELIST,    ELEMLIST,    FREQLIST,    TIMELIST,
1!            ITERLIST,    GDVLIST,     LDVLIST,     DCONLIST,    PLYLIST;
1!$
1!RELATION    GPFELEM,     EOSUMMRY,    EOBAR,       EOELAS,      EOHEX1,
1!            EOHEX2,      EOHEX3,      EOQDMM1,     EOQUAD4,     EOROD,
1!            EOSHEAR,     EOTRMEM,     GPFDATA,     EOTRIA3;
1!UNSTRUCT    EODISC;
1!$
1!RELATION    OGRIDLOD,    OGRIDDSP,    OLOCALDV,    OAGRDDSP,    OAGRDLOD;
1!MATRIX      [FLUTMODE],  [PTGLOAD],   [PFGLOAD],   [PTHLOAD],   [PFHLOAD];
1!$
1!$*************************************************************************
1!$          DECLARATIONS FOR MODULES EMA1, EMA2 AND GLOBAL
1!$                    MATRIX PARTITION/REDUCTION
1!$*************************************************************************
1!$
1!IUNSTRUCT   DKVI,        DMVI;
1!RELATION    GMKCT,       GMMCT;
1!MATRIX      [KGG],       [KNN],       [KFF],       [KAA],       [KLL],
1!            [MGG],       [MNN],       [MFF],       [MAA],       [MLL],
1!            [MRRBAR],    [MLR],       [KFS],       [KSS],       [KOOINV(1000)],
1!            [GSUBO(1000)],            [KLLINV(1000)],           [MRR(1000)],
1!            [IFM(1000)], [M1GG],      [IFR(1000)], [KRR],       [D(1000)],
1!            [KLR],       [K1GG],      [LHS(1000)], [M2GG],      [MOO],
1!            [MOA],       [K2GG],      [MAABAR];
1!MATRIX      [TMP1],      [TMP2];
1!MATRIX      [PG],        [PN],        [PF],        [PA],
1!            [PO],        [PLBAR],     [PR],        [RHS(1000)], [UG(1000)],
1!            [UN],        [UF],        [UA],        [UL],        [UM],
1!            [AG(1000)],  [AN],        [AF],        [AA],        [AR],
1!            [AL],        [UO],        [UOO],       [PS];
1!LOGICAL     M2GGFLAG,    K2GGFLAG;
1!$
1!$*************************************************************************
1!$      DECLARATIONS FOR SOLUTION CONTROL
1!$*************************************************************************
1!$
1!INTEGER     NUMOPTBC,    NBNDCOND,    MAXITER,
1!            MPS,         MPE,
1!            OCS,         OCE,
1!            FSDS,        FSDE;
1!INTEGER     BLOAD,       BMASS,       BMODES,      BSAERO,      BFLUTR,
1!            BDYN,        BDRSP,       BDTR,        BMTR,        BDFR,
1!            BMFR,        BGUST,       BBLAST,      NMPC,        NSPC,
1!            NOMIT,       NRSET,       DMODES;
1!REAL        MOVLIM,      WINDOW,      OCMOVLIM,    ALPHA,       CNVRGLIM,
1!            NRFAC,       EPS;
1!RELATION    JOB,         OPTIMIZE,    CASE;
1!$
1!$*************************************************************************
1!$      DECLARATIONS FOR SENSITIVITY  EVALUATION
1!$*************************************************************************
1!$
1!INTEGER     DDFLG,       NACSD,       NAUS,        NAUA;
1!LOGICAL     ACTBOUND,    ACTFLUT,     ACTDYN,      ACTAERO,     ACTAEFF,
1!            ACTUAG,      ACTUAGG,     ACTPNL,      ACTBAR;
1!UNSTRUCT    PCAS,        PCAA,        PCAE;
1!RELATION    PDLIST;
1!MATRIX      [DFDU],      [PGAS],      [UGA],       [DUG],       [DMUG],
1!            [DPFV],      [DPOV],      [DPNV],      [DPAV],      [DUAV],
1!            [DUAD],      [DUFV],      [AGA],       [AMAT],      [DKUG],
1!            [DPGV],      [DPLV],      [DURD],      [DULD],      [DULV],
1!            [DDELDV],    [DPRV],      [DRHS],      [DFDUF],     [PGAA],
1!            [DFDUN],     [DMAG],      [DMUN],      [DMUF],      [DMUA],
1!            [DMUO],      [DMUL],      [DMUR],      [DMU],       [DP1],
1!            [DK1V],      [AUAGC],     [DURV],      [EFFSENS],   [DU1L],
1!            [DU1R],      [DU2],       [LHSL],      [LHSU],      [PGAU],
```

```
146   1!          [SENSMT];                                                                    !
147   1!IMATRIX   [GLBSIG],    [DPTHVI],    [DPGRVI],    [DPVJ];                             !
148   1!$                                                                                    $!
149   1!$*************************************************************************************$!
150   1!$     AERODYNAMIC ENTITIES                                                           $!
151   1!$*************************************************************************************$!
152   1!$                                                                                    $!
153   1!INTEGER   SYM,         MINDEX,      SUB,         S;                                  !
154   1!REAL      QDP,         MACH;                                                         !
155   1!LOGICAL   LOOP,        AEFLG(1000), NONPONLY;                                       !
156   1!UNSTRUCT  ACPT,        UNMK;                                                         !
157   1!RELATION  AESURF,      AIRFOIL,     AEROS,       AEFACT,      AXSTA,               !
158   1!          BODY,        SPLINE1,     SET1,        SET2,        ATTACH,              !
159   1!          TRIM,        AERO,        BLAST,       CAERO6,      PAERO6,              !
160   1!          GEOMSA,      AECOMPS,     STABCF,      CAERO1,      PAERO1,              !
161   1!          CAERO2,      PAERO2,      MKAERO1,     MKAERO2,     FLUTTER,             !
162   1!          FLFACT,      CLAMBDA,     CONEFFS,     CONLINK,     GEOMUA,              !
163   1!          AECOMPU,     SPLINE2,     CONEFFF,     AEROGEOM,    CAROGEOM,            !
164   1!          AERUGEOM,    CAROUGEO,    ACOORD,      AGRID,       AGRIDZ,              ! ←
165   1!          AQUAD4,      ATRIA3,      CAERO7,      PAFOIL7,     BODY7,               !
166   1!          PBODY7,      SEGMESH,     CHORDCP,     MACHCP,      ZTAIC,               !
167   1!          AECOMPZ,     GEOMZA,      MKAERO2,     AERO2,       REUNMK,              !
168   1!          PANLST1,     PANLST2,     SPLINE3,     AESURFZ,     TRIMFLT,             !
169   1!          LOADMOD;                                                                  ! ←
170   1!MATRIX    [AIRFRC(1000)],          [AICMAT(1000)],          [AAICMAT(1000)],        !
171   1!          [AICS],      [KAFF],     [PAF],       [KAAA],      [PAA],               !
172   1!          [GASUBO(30,33)],         [SKJ],       [D1JK],      [D2JK],              !
173   1!          [KARL],      [R11],      [K21(30,33)],[PARBAR],    [PAL],               !
174   1!          [PAR(30,33)],[K1112(30,33)],          [AIRFORCE],  [K22],               !
175   1!          [GTKG],      [GTKN],     [GTKF],      [GSTKG],     [GSTKN],             !
176   1!          [GSTKF],     [GSKF],     [UGTKG],     [UGTKN],     [UGTKF],             !
177   1!          [UGTKA],     [UGTKO],    [UGTKAB],    [AITD],      [KARR],              !
178   1!          [R12(30,33)],[R22],      [R32(30,33)],[K11],       [K12(30,33)],        !
179   1!          [P1],        [R21(30,33)],[R31(30,33)],[RL11(30,33)],                    !
180   1!          [RU11(30,33)],           [P2],        [MAAA],      [IFMA(30,33)],       !
181   1!          [R13(30,33)],[R33],      [DELC],      [PRIGID],                         !
182   1!          [AARC],      [AAR],      [AAA(1000)], [UAA(1000)], [AAAGC],             !
183   1!          [PAO(1000)], [AAFTMP],   [UAFTMP],    [UAN],       [AAN],               !
184   1!          [UAG(1000)], [AAG(1000)],[AAL],       [AAF],       [UAF],               !
185   1!          [KOOL(30,33)],           [KOOU(30,33)],           [LHSA(30,33)],       !
186   1!          [POARO(30,33)],          [KAO(30,33)],[UAR],      [RHSA(30,33)],       !
187   1!          [DELTA(1000)],           [PAOC(1000)],[UAAC(1000)],[AAAC(1000)],       !
188   1!          [UAFC(1000)],[UANC(1000)],[UAGC(30,33)],          [AAFC(1000)],        !
189   1!          [AANC(1000)],[AAGC(30,33)],          [KL11(30,33)],[KU11(30,33)],     !
190   1!          [R11DPL],    [R11PAL(30,33)],        [R1112(30,33)],                   !
191   1!          [R1113(30,33)],          [UAL];                                         !
192   1!IMATRIX   [AJJTL],     [QJJL],     [QKKL],      [QHHL],      [AJK],               ! ←
193   1!          [AJC],       [SCNTLG],   [SCNTLK],    [ACNTLK],    [ACNTLG],            !
194   1!          [QGK],       [LMODEG],   [LMODEK],    [AJL];                            ! ←
195   1!$                                                                                    $!
196   1!$*************************************************************************************$!
197   1!$     DYNAMIC RESPONSE DECLARATIONS                                                  $!
198   1!$*************************************************************************************$!
199   1!$                                                                                    $!
200   1!INTEGER   HSIZE(1000);                                                              !
201   1!UNSTRUCT  TFDATA,      ICDATA,      UDLOLY;                                         !
202   1!RELATION  LAMBDA,      OEIGS,       DLONLY,      DLOAD,       TABLED1,             !
203   1!          IC,          TLOAD1,      TLOAD2,      RLOAD1,      RLOAD2,              !
204   1!          TSTEP,       VSDAMP,      TABDMP1,     DLAGS,       TF,                  !
205   1!          DMIG,        GUST,        FREQ,        FREQ1,       FREQ2,               !
206   1!          FFT,         FLUTREL;                                                     !
207   1!MATRIX    [PHIKH],     [QHJL],      [QKJL],      [PHIA],      [MII],               !
208   1!          [PHIO],      [PHIF],      [PHIN],      [PHIG(1000)],[KHHT],              !
209   1!          [KHHF],      [BHH],       [MHH],       [PDT],       [PDF],               !
210   1!          [KDDT],      [KDDF],      [BDD],       [MDD],       [ICMATRIX],          !
211   1!          [UTRANA],    [UFREQA],    [UTRANI],    [UFREQI],    [UFREQE],            !
212   1!          [UTRANE],    [UTRANF],    [UFREQF],    [UTRANN],    [UFREQN],            !
213   1!          [UTRANG],    [UFREQG],    [MHHFL(30,33)],           [BHHFL(30,33)],      !
214   1!          [QHHLFL(30,33)],          [KHHFL(30,33)];                                !
215   1!$                                                                                    $!
216   1!$*************************************************************************************$!
217   1!$  DECLARATIONS FOR GENERALIZED DYNAMIC REDUCTION (GDR)                              $!
218   1!$*************************************************************************************$!
219   1!$                                                                                    $!
220   1!INTEGER   LKSET,       LJSET,       NEIV,        GNORM,       NGDR,                !
221   1!          ASIZE,       LSIZE;                                                       !
222   1!REAL      FMAX;                                                                     !
223   1!RELATION  DYNRED;                                                                   !
224   1!MATRIX    [PGDRG(1000)],[PHIOK],    [KOO],       [GGO],       [KSOO],              !
225   1!          [KOA],       [LSOO],      [PAJK],      [PFJK],      [UFGDR],             !
226   1!          [AFGDR],     [UJK],       [GTMP];                                       !
```

```
227   1!$                                                                                $!
228   1!$*********************************************************************************$!
229   1!$     BLAST RESPONSE DECLARATIONS                                                $!
230   1!$*********************************************************************************$!
231   1!$                                                                                $!
232   1!REAL       BQDP;                                                                  !
233   1!MATRIX     [MPART],     [ID2],       [PHIE],      [PHIR],      [PHIB],          !
234   1!           [GENM],      [GENK],      [GENF],      [GENQ],      [GENQL],         !
235   1!           [DTSLP],     [FTF],       [QRE],       [QEE],       [KEQE],          !
236   1!           [LKQ],       [UKQ],       [GFR],       [GFE],       [BTEM],          !
237   1!           [BLSTJA],    [BLGTJA],    [BFRC],      [MATTR],     [MATSS],         !
238   1!           [KEE],       [DELB],      [DELM],      [URDB],      [GENFA],         !
239   1!           [DWNWSH],    [ELAS],      [SLPMOD],    [QRR],       [UBLASTI],       !
240   1!           [UBLASTG],   [UBLASTF];                                               !
241   1!$                                                                                $!
242   1!$*********************************************************************************$!
243   1!$                                                                                $!
244   1!$                BEGIN MAPOL SOLUTION SEQUENCE                                   $!
245   1!$                                                                                $!
246   1!$*********************************************************************************$!
247   1!$     PREFACE MODULES                                                            $!
248   1!$*********************************************************************************$!
249   1!SINGOSET := 1;                                                                    !
250   1!SINGASET := 2;                                                                    !
251   1!SINGLSET := 3;                                                                    !
252   1!$*********************************************************************************$!
253   1!$                                                                                $!
254   1!$     INITIALIZE SUBSCRIPT VALUES TO "1" TO AVOID RUN TIME PROBLEMS              $!
255   1!$                                                                                $!
256   1!$*********************************************************************************$!
257   1!SUB      := 1;                                                                    !
258   1!PRINT("LOG=('BEGIN PREFACE MODULES')");                                           !
259   1!CALL SOLUTION ( NUMOPTBC, NBNDCOND, MPS, MPE, OCS, OCE, FSDS, FSDE,              !
260   1!                MAXITER, MOVLIM, WINDOW, OCMOVLIM, ALPHA, CNVRGLIM,               !
261   1!                NRFAC, EPS );                                                     !
262   1!CALL IFP ( GSIZEB );                                                              !
263   1!$*********************************************************************************$!
264   1!$ TRY USING A UTILITY TO PRINT OUT THE GRID RELATIONAL ENTITY                     $!
265   1!$                                                                                $!
266   1!$             GENERATE THE ELEMENT MATRICES                                      $!
267   1!$                                                                                $!
268   1!$*********************************************************************************$!
269   1!PRINT("LOG=('ELEMENT MATRIX GENERATION')");                                       !
270   1!$*********************************************************************************$!
271   1!$                                                                                $!
272   1!$*********************************************************************************$!
273   1!CALL MAKEST ( NDV, GLBDES, [PTRANS], [PMINT], [PMAXT], LOCLVAR,                  !
274   1!              TFIXED, DESLINK );                                                  !
275   1!$*********************************************************************************$!
276   1!$                                                                                $!
277   1!$*********************************************************************************$!
278   1!CALL EMG ( NDV, GSIZEB, GLBDES, DESLINK, [SMAT], DVCT, DVSIZE,                   !
279   1!           KELM, MELM, TELM, TREF );                                              !
280   1!CALL PFBULK ( GSIZEB, EOSUMMRY, EODISC, GPFELEM );                               !
281   1!$*********************************************************************************$!
282   1!$             HANDLE THE NON-PLANAR STEADY AERODYNAMICS ANALYSES                 $!
283   1!$             TERMINATE THE EXECUTION IF THE ONLY DISCIPLINE IS NPSAERO           $!
284   1!$                                                                                $!
285   1!$PRINT("LOG=('NON-PLANAR STEADY AERODYNAMICS')");                                $! ←
286   1!$CALL STEADYNP ( NONPONLY, AECOMPS, GEOMSA, STABCF, [AIRFORCE], AEROGEOM,        $!
287   1!$              CAROGEOM, OAGRDLOD );                                             $! ←
288   1!$IF NONPONLY CALL EXIT;                                                          $!
289   1!$                                                                                $!
290   1!$             ASSEMBLE THE ELEMENT MATRICES                                      $!
291   1!$             TO THE SENSITIVITY MATRICES                                        $!
292   1!$                                                                                $!
293   1!$*********************************************************************************$!
294   1!PRINT("LOG=('PHASE 1 ELEM. MATRIX ASSEMBLY')");                                   !
295   1!CALL EMA1  ( NDV, GLBDES, DVCT, KELM, MELM, GMKCT, DKVI, GMMCT, DMVI );          !
296   1!$*********************************************************************************$!
297   1!$             GENERATE THE SIMPLE LOAD VECTORS                                   $!
298   1!$             AND LOAD SENSITIVITIES                                             $!
299   1!$                                                                                $!
300   1!$*********************************************************************************$!
301   1!PRINT("LOG=('PHASE 1 STATIC LOADS GENER.')");                                     !
302   1!CALL LODGEN ( GSIZEB, GLBDES, DVCT, DVSIZE, GMMCT, DMVI, TELM, TREF,             !
303   1!              SMPLOD, [DPTHVI], [DPGRVI] );                                      !
304   1!$*********************************************************************************$!
305   1!$                                                                                $!
306   1!$             GENERATE THE STEADY AIC MATRIX AND THE                             $!
307   1!$             STEADY SPLINE TRANSFORMATION MATRICES                              $!
```

```
1!$                                                                          $!
1!$PRINT("LOG=('STEADY AERODYNAMICS')");                                     $!
1!$LOOP   := TRUE;                                                           $!
1!$MINDEX := 0;                                                              $!
1!$WHILE LOOP DO                                                             $!
1!$   MINDEX := MINDEX + 1;                                                  $!
1!$   CALL STEADY ( MINDEX, LOOP, AECOMPS, GEOMSA, STABCF, [AICMAT(MINDEX)],  $!
1!$                 [AAICMAT(MINDEX)], [AIRFRC(MINDEX)], AEROGEOM, CAROGEOM );  $!
1!$ENDDO;                                                                    $!
1!$CALL SPLINES ( GSIZEB, GEOMSA, AECOMPS, AEROS, [GTKG], [GSTKG] );         $!
1!$                                                                          $!
1!$             GENERATE THE UNSTEADY AIC MATRIX AND THE                     $!
1!$             UNSTEADY SPLINE TRANSFORMATION MATRIX                        $!
1!$                                                                          $!
1!$PRINT("LOG=('UNSTEADY AERODYNAMICS')");                                   $!
1!$CALL UNSTEADY ( GEOMUA, AECOMPU, [AJJTL], [D1JK], [D2JK], [SKJ],          $!
1!$                AERUGEOM, CAROUGEO );                                     $!
1!$CALL AMP ( [AJJTL], [D1JK], [D2JK], [SKJ], [QKKL], [QKJL], [QJJL] );      $!
1!$CALL SPLINEU ( GSIZEB, GEOMUA, AECOMPU, AERO, [UGTKG] );                  $!
1!$*************************************************************************$!
1!$                                                                          $!
1!$             ZAERO MODULE   P. C. CHEN 3-28-1997                          $!
1!$                                                                          $!
1!$*************************************************************************$!
1!$*************************************************************************$!
1!PRINT("LOG=('ZAERO AERODYNAMIC GEOMETRY')");                                !
1!PRINT("LOG=('ZAERO AERODYNAMIC GEOMETRY')");                                !
1!$*************************************************************************$!
1!$ CALL AEROGM MODULE                                                       $!
1!$ FOR BOTH STEADY AND UNSTEADY GEOMETRY GENERATIONS                        $!
1!$*************************************************************************$!
1!CALL AEROGM ( AECOMPZ, GEOMZA, AGRIDZ );                                   !
1!$*************************************************************************$!
1!PRINT("LOG=('ZAERO CONTROL MODE MODULE ')");                                !
1!PRINT("LOG=('ZAERO CONTROL MODE MODULE ')");                                !
1!$*************************************************************************$!
1!CALL CONMOD ( GEOMZA,AECOMPZ,[SCNTLG],[SCNTLK],[ACNTLG],[ACNTLK],[LMODEG],  !
1!              [LMODEK] );                                                   !
1!$*************************************************************************$!
1!PRINT("LOG=('ZAERO SPLINE MODULE ')");                                      !
1!PRINT("LOG=('ZAERO SPLINE MODULE ')");                                      !
1!$*************************************************************************$!
1!CALL SPLINZ  ( GSIZEB, GEOMZA, AECOMPZ, AEROZ, [UGTKG] );                  !
1!$*************************************************************************$!
1!$*************************************************************************$!
1!$ CALL ZAEROM MODULE                                                       $!
1!$ FOR BOTH STEADY AND UNSTEADY AIC GENERATIONS                             $!
1!$*************************************************************************$!
1!PRINT("LOG=('ZAERO UNSTEADY AERODYNAMICS ')");                              !
1!PRINT("LOG=('ZAERO UNSTEADY AERODYNAMICS ')");                              !
1!CALL UZAERO ( AECOMPZ, GEOMZA, AGRIDZ, [AJK], [AJC], [AJL], [QGK],         !
1!              [SKJ], [SCNTLK], [ACNTLK], [LMODEK]);                        !
1!PRINT("LOG=('ZAERO STEADY AERODYNAMICS')");                                 !
1!PRINT("LOG=('ZAERO STEADY AERODYNAMICS')");                                 !
1!LOOP   := TRUE;                                                            !
1!MINDEX := 0;                                                               !
1!WHILE LOOP DO                                                              !
2!   MINDEX := MINDEX + 1;                                                   !
2!   CALL SZAERO ( [AJK], MINDEX, LOOP, AECOMPZ, GEOMZA, AGRIDZ, STABCF,     !
2!                 [AICMAT(MINDEX)], [AAICMAT(MINDEX)], [AIRFRC(MINDEX)],    !
2!                 [SCNTLK], [ACNTLK] );                                     !
2!ENDDO;                                                                     !
1!$*************************************************************************$!
1!$                                                                          $!
1!$*************************************************************************$!
1!$        BEGIN OPTIMIZATION LOOP                                           $!
1!$*************************************************************************$!
1!$                                                                          $!
1!IF NUMOPTBC > 0 THEN                                                       !
2!   PRINT("LOG=('*****************************')");                         !
2!   PRINT("LOG=('BEGIN OPTIMIZATION')");                                    !
2!$                                                                         $!
2!$  INITIALIZE MAPOL PARAMETERS                                            $!
2!$                                                                         $!
2!   GLBCNVRG := FALSE;                                                      !
2!   APPCNVRG := FALSE;                                                      !
2!$                                                                         $!
2!$  BEGIN CONVERGENCE LOOP                                                 $!
2!$                                                                         $!
2!   WHILE NOT GLBCNVRG AND NITER <= MAXITER DO                              !
3!$                                                                         $!
```

```
389    3!$     ASSEMBLE THE GLOBAL MATRICES                                                    $!
390    3!$                                                                                     $!
391    3!      NITER := NITER + 1;                                                              !
392    3!      PRINT("LOG=('----------------------------')");                                   !
393    3!      PRINT("LOG=('   DESIGN ITERATION ',I3)",NITER);                                  !
394    3!      CALL ITERINIT ( NITER, CONST );                                                  !
395    3!      CALL UTMPRG ( [GLBSIG] );                                                        !
396    3!      CALL TCEVAL ( NITER, NDV, MOVLIM, WINDOW, GLBDES, LOCLVAR, [PMINT],              !
397    3!                    [PMAXT], TFIXED, CONST );                                          !
398    3!      CALL LAMINCON ( NITER, NDV, DCONLAM, DCONLMN, DCONPMN, TFIXED, GLBDES,           !
399    3!                      LOCLVAR, [PTRANS], CONST );                                      !
400    3!      CALL EMA2 ( NITER, NDV, GSIZEB, GLBDES, GMKCT, DKVI, [K1GG],                     !
401    3!                                              GMMCT, DMVI, [M1GG] );                   !
402    3!$                                                                                    $!
403    3!$     BEGIN BOUNDARY CONDITION LOOP FOR OPTIMIZATION                                 $!
404    3!$                                                                                    $!
405    3!      FOR BC = 1 TO NUMOPTBC DO                                                        !
406    4!         PRINT("LOG=('       BOUNDARY CONDITION ',I3)",BC);                          !
407    4!$                                                                                    $!
408    4!$        ESTABLISH THE BASE USET AND PARTITIONING DATA FOR THE BC                     $!
409    4!$        THIS DATA MUST BE RECREATED EACH ITERATION SINCE GDR CAN CHANGE IT           $!
410    4!$                                                                                    $!
411    4!         CALL MKUSET( BC, GSIZEB, [YS(BC)], [TMN(BC)], [PGMN(BC)], [PNSF(BC)],        !
412    4!                      [PFOA(BC)], [PARL(BC)], USET(BC) );                           !
413    4!$                                                                                    $!
414    4!$        MAKE B.C.-DEPENDENT BGPDT FROM BASE, ADDING THE EXTRA POINTS FOR             $!
415    4!$        THIS B.C.                                                                   $!
416    4!$                                                                                    $!
417    4!         CALL BCBGPDT( BC , GSIZEB , BGPDT(BC) , ESIZE(BC) );                         !
418    4!         GSIZE      := GSIZEB;                                                        !
419    4!         PSIZE(BC) := ESIZE(BC) + GSIZE;                                              !
420    4!$                                                                                    $!
421    4!$        PROCESS MATRICES, TRANSFER FUNCTIONS, AND INITIAL CONDITIONS FOR             $!
422    4!$        THIS B.C.                                                                   $!
423    4!$                                                                                    $!
424    4!         CALL BCBULK( BC , PSIZE(BC) , BGPDT(BC) , USET(BC) );                        !
425    4!$                                                                                    $!
426    4!         CALL BOUND  ( BC, GSIZE, ESIZE(BC), USET(BC), BLOAD, BMASS, DMODES,          !
427    4!                       BMODES, BSAERO, BFLUTR, BDYN, BDRSP, BDTR, BMTR, BDFR,       !
428    4!                       BMFR, BGUST, BBLAST, NMPC, NSPC, NOMIT, NRSET, NGDR );       !
429    4!$                                                                                    $!
430    4!$        DETERMINE IF ANY M2GG/K2GG INPUT DATA ARE TO BE ADDED                       $!
431    4!$                                                                                    $!
432    4!         CALL NULLMAT ( [KGG], [MGG] );                                               !
433    4!         CALL MK2GG ( BC, GSIZEB, [M2GG], M2GGFLAG, [K2GG], K2GGFLAG );               !
434    4!         IF M2GGFLAG THEN                                                             !
435    5!            [MGG] := [M1GG] + [M2GG];                                                 !
436    5!         ELSE                                                                         !
437    5!            [MGG] := [M1GG];                                                          !
438    5!         ENDIF;                                                                       !
439    4!         IF K2GGFLAG THEN                                                             !
440    5!            [KGG] := [K1GG] + [K2GG];                                                 !
441    5!         ELSE                                                                         !
442    5!            [KGG] := [K1GG];                                                          !
443    5!         ENDIF;                                                                       !
444    4!$                                                                                    $!
445    4!$        CALL THE GRID POINT WEIGHT GENERATOR FOR THIS BOUNDARY CONDITON             $!
446    4!$                                                                                    $!
447    4!         CALL GPWG ( NITER, BC, GPWGGRID, [MGG], OGPWG );                            !
448    4!$                                                                                    $!
449    4!         IF BLOAD <> 0 CALL GTLOAD (NITER, BC, GSIZE, BGPDT(BC), GLBDES,              !
450    5!                                    SMPLOD, [DPTHVI], [DPGRVI], [PG], OGRIDLOD);!
451    4!$                                                                                    $!
452    4!$        PARTITION-REDUCTION OF GLOBAL MATRICES                                      $!
453    4!$                                                                                    $!
454    4!$******************************* TAKEN OUT FOR ZAERO ***************************$! ←
455    4!$        IF NUMOPTBC > 1  CALL NULLMAT ( [KNN], [PN], [MNN],                         $!
456    4!$                                        [GTKN], [GSTKN], [UGTKN] );                 $!
457    4!$*************************************************************************************$!
458    4!         IF NUMOPTBC > 1  CALL NULLMAT ( [KNN], [PN], [MNN], [UGTKN] );               ! ←
459    4!         IF NMPC <> 0 THEN                                                            !
460    5!$                                                                                    $!
461    5!$           PERFORM MPC REDUCTION                                                    $!
462    5!$                                                                                    $!
463    5!            PRINT("LOG=('         MPC REDUCTION')");                                  !
464    5!            CALL GREDUCE ( [KGG], [PG], [PGMN(BC)], [TMN(BC)], [KNN], [PN] );       !
465    5!            IF BMASS <> 0  CALL GREDUCE ( [MGG],,[PGMN(BC)], [TMN(BC)], [MNN] );!
466    5!$******************************* TAKEN OUT FOR ZAERO ***************************$! ←
467    5!$           IF BSAERO <> 0 THEN                                                      $!
468    5!$              CALL GREDUCE (, [GTKG], [PGMN(BC)], [TMN(BC)], , [GTKN]);            $!
469    5!$              CALL GREDUCE (, [GSTKG], [PGMN(BC)], [TMN(BC)], , [GSTKN]);          $!
```

```
5!$          ENDIF;                                                                    $!
5!$*************************************************************************************$!
5!           IF BFLUTR <> 0 OR BGUST <> 0 OR BBLAST <> 0 OR BSAERO <> 0                 ! ←
6!              CALL GREDUCE (, [UGTKG], [PGMN(BC)], [TMN(BC)], ,[UGTKN] );            !
5!        ELSE                                                                         !
5!$                                                                                    $!
5!$          NO MPC REDUCTION                                                         $!
5!$                                                                                    $!
5!           [KNN] := [KGG];                                                          !
5!           IF BLOAD <> 0 [PN] := [PG];                                              !
5!           IF BMASS <> 0 [MNN] := [MGG];                                            !
5!$****************************** TAKEN OUT FOR ZAERO ***************************$! ←
5!$          IF BSAERO <> 0 THEN                                                      $!
5!$             [GTKN]  := [GTKG];                                                    $!
5!$             [GSTKN] := [GSTKG];                                                   $!
5!$          ENDIF;                                                                   $!
5!$*************************************************************************************$!
5!           IF BFLUTR <> 0 OR BGUST <> 0 OR BBLAST <> 0 OR BSAERO <>0                !
6!              [UGTKN]  := [UGTKG];                                                  ! ←
5!        ENDIF;                                                                       !
4!$                                                                                   $!
4!$       PERFORM AUTOSPC CALCULATIONS ON THE KNN MATRIX                              $!
4!$                                                                                   $!
4!        PRINT("LOG=('         AUTOSPC COMPUTATIONS')");                            !
4!        CALL GPSP ( NITER, BC, NGDR, [KNN], BGPDT(BC), [YS(BC)],                   !
4!                    USET(BC), GPST(BC) );                                          !
4!        CALL MKPVECT ( USET(BC), [PGMN(BC)], [PNSF(BC)],                           !
4!                                 [PFOA(BC)], [PARL(BC)] );                         !
4!        CALL BOUNDUPD ( BC, GSIZE, ESIZE(BC), USET(BC), NSPC, NOMIT, NRSET );      !
4!$                                                                                   $!
4!$       FOR SENSITIVITY ANALYSIS, SAVE A COPY OF THE PRE-GDR PART. VECTS.           $!
4!$                                                                                   $!
4!        CALL MKPVECT ( USET(BC), [PGMNS(BC)], [PNSFS(BC)],                         !
4!                                 [PFOAS(BC)], [PARLS(BC)] );                       !
4!$                                                                                   $!
4!$****************************** TAKEN OUT FOR ZAERO ***************************$! ←
4!$       IF NUMOPTBC > 1 CALL NULLMAT ( [KFF], [PF], [MFF], [GTKF], [GSTKF],        $!
4!$                                      [UGTKF] );                                   $!
4!$*************************************************************************************$!
4!        IF NUMOPTBC > 1 CALL NULLMAT ( [KFF], [PF], [MFF], [UGTKF] );             ! ←
4!        IF NSPC <> 0 THEN                                                          !
5!$                                                                                   $!
5!$          PERFORM SPC REDUCTION                                                    $!
5!$                                                                                   $!
5!           PRINT("LOG=('         SPC REDUCTION')");                               !
5!           CALL NREDUCE ( [KNN], [PN], [PNSF(BC)], [YS(BC)], [KFF], [KFS],        !
5!                          [KSS], [PF], [PS] );                                     !
5!           IF BMASS <> 0 CALL NREDUCE ( [MNN], , [PNSF(BC)], , [MFF] );           !
5!$****************************** TAKEN OUT FOR ZAERO ***************************$! ←
5!$          IF BSAERO <> 0 THEN                                                      $!
5!$             CALL NREDUCE ( , [GTKN], [PNSF(BC)], , , , , [GTKF] );                $!
5!$             CALL NREDUCE ( , [GSTKN], [PNSF(BC)], , , , , [GSTKF] );              $!
5!$          ENDIF;                                                                   $!
5!$*************************************************************************************$!
5!           IF BFLUTR <> 0 OR BGUST <> 0 OR BBLAST <> 0 OR BSAERO <> 0               ! ←
6!              CALL NREDUCE (,[UGTKN], [PNSF(BC)],,,,, [UGTKF]);                     !
5!        ELSE                                                                       !
5!$                                                                                   $!
5!$          NO SPC REDUCTION                                                         $!
5!$                                                                                   $!
5!           [KFF] := [KNN];                                                         !
5!           IF BLOAD <> 0 [PF] := [PN];                                             !
5!           IF BMASS <> 0 [MFF] := [MNN];                                           !
5!$****************************** TAKEN OUT FOR ZAERO ***************************$! ←
5!$          IF BSAERO <> 0 THEN                                                      $!
5!$             [GTKF]  := [GTKN];                                                    $!
5!$             [GSTKF] := [GSTKN];                                                   $!
5!$          ENDIF;                                                                   $!
5!$*************************************************************************************$!
5!           IF BFLUTR <> 0 OR BGUST <> 0 OR BBLAST <> 0 OR BSAERO <> 0               !
6!              [UGTKF]  := [UGTKN];                                                  ! ←
5!        ENDIF;                                                                     !
4!$                                                                                 $!
4!        IF NUMOPTBC > 1 CALL NULLMAT ( [KAA], [PA], [MAA],                       !
5!                                       [KAAA], [PAA], [UGTKA] );                  !
4!$                                                                                 $!
4!        IF NGDR <> 0 THEN                                                        !
5!$                                                                                 $!
5!$          PERFORM THE GENERAL DYNAMIC REDUCTION WHICH IS DISCIPLINE               $!
5!$          INDEPENDENT.  THE RESULTING [GSUBO] MATRIX WILL BE USED BY             $!
5!$          ALL DISCIPLINES                                                        $!
```

```
551    5!$                                                                    $!
552    5!          PRINT("LOG=('         DYNAMIC REDUCTION')");                !
553    5!$                                                                    $!
554    5!$         OBTAIN THE OMITTED DOF PARTITION OF KFF AND MFF            $!
555    5!$                                                                    $!
556    5!          CALL PARTN ( [KFF], [KOO], , [KOA], , [PFOA(BC)] );        !
557    5!          CALL PARTN ( [MFF], [MOO], , , , [PFOA(BC)] );             !
558    5!          ASIZE := GSIZE - NMPC - NSPC - NOMIT;                      !
559    5!          LSIZE := ASIZE - NRSET;                                    !
560    5!          CALL GDR1 ( [KOO], [MOO], [KSOO], [GGO], LKSET, LJSET, NEIV,  !
561    5!                      FMAX, BC, BGPDT(BC), USET(BC), NOMIT, LSIZE );  !
562    5!$                                                                    $!
563    5!$         LKSET              MEANING                                 $!
564    5!$               <> 0         APPROX. MODE SHAPES SELECTED            $!
565    5!$                = 0         NO APPROX. MODE SHAPES IN GDR           $!
566    5!$                                                                    $!
567    5!          IF LKSET <> 0  THEN                                        !
568    6!             CALL SDCOMP ( [KSOO], [LSOO], USET(BC), SINGOSET );     !
569    6!             CALL GDR2   ( [LSOO], [MOO], [PHIOK], LKSET, LJSET,     !
570    6!                           NEIV,   FMAX,  BC );                      !
571    6!          ENDIF;                                                     !
572    5!          CALL GDR3 ( [KOO], [KOA], [MGG], [PHIOK], [TMN(BC)], [GGO],   !
573    5!                      [PGMN(BC)], [PNSF(BC)], [PFOA(BC)], [GSUBO(BC)],   !
574    5!                      BGPDT(BC), USET(BC),                           !
575    5!                      LKSET, LJSET, ASIZE, GNORM, BC );              !
576    5!          CALL GDR4 ( BC, GSIZE, PSIZE(BC), LKSET, LJSET, NUMOPTBC, NBNDCOND,  !
577    5!                      [PGMN(BC)], [TMN(BC)], [PNSF(BC)], [PFOA(BC)],   !
578    5!                      [PARL(BC)], [PGDRG(BC)], [PAJK], [PFJK], BGPDT(BC),  !
579    5!                      USET(BC) );                                    !
580    5!       ENDIF;                                                       $!
581    4!$                                                                    !
582    4!       IF BLOAD <> 0 OR BMODES <> 0 OR BFLUTR <> 0 OR BDYN <> 0 THEN $!
583    5!$                                                                    $!
584    5!$         REDUCE THE MATRICES WITHOUT AEROELASTIC CORRECTIONS         $!
585    5!$                                                                    $!
586    5!          IF NGDR <> 0 THEN                                          !
587    6!$                                                                    $!
588    6!$            PERFORM THE GENERAL DYNAMIC REDUCTION                   $!
589    6!$                                                                    $!
590    6!             PRINT("LOG=('         SYMMETRIC DYNAMIC REDUCTION')");  !
591    6!$                                                                    $!
592    6!             [MAA] := TRANS ( [GSUBO(BC)] ) * [ [MFF] * [GSUBO(BC)] ];  !
593    6!             [KAA] := TRANS ( [GSUBO(BC)] ) * [ [KFF] * [GSUBO(BC)] ];  !
594    6!             IF BLOAD <> 0  [PA]    := TRANS ( [GSUBO(BC)] ) * [PF];    !
595    6!             IF BFLUTR <> 0 OR BGUST <> 0 OR BBLAST <> 0 THEN           !
596    7!                [TMP1]  := TRANS ( [UGTKF] ) * [GSUBO(BC)];             !
597    7!                CALL TRNSPOSE ( [TMP1], [UGTKA] );                      !
598    7!             ENDIF;                                                     !
599    6!          ELSE                                                       !
600    6!             IF NOMIT <> 0 THEN                                      !
601    7!$                                                                    $!
602    7!$               PERFORM THE STATIC REDUCTION                         $!
603    7!$                                                                    $!
604    7!                PRINT("LOG=('         STATIC CONDENSATION')");       !
605    7!$                                                                    $!
606    7!                CALL FREDUCE ( [KFF], [PF], [PFOA(BC)], , [KOOINV(BC)], , ,  !
607    7!                               [GSUBO(BC)], [KAA], [PA], [PO], USET(BC) );  !
608    7!$                                                                    $!
609    7!                IF BMASS <> 0 THEN                                   !
610    8!$                                                                    $!
611    8!$                  PERFORM GUYAN REDUCTION OF THE MASS MATRIX         $!
612    8!$                                                                    $!
613    8!                   CALL PARTN ( [MFF], [MOO], , [MOA], [MAABAR], [PFOA(BC)] );!
614    8!                   [MAA] := [MAABAR] + TRANS([MOA]) * [GSUBO(BC)] +     !
615    8!                            TRANS([GSUBO(BC)]) * [MOA] +                !
616    8!                            TRANS([GSUBO(BC)]) * [ [MOO] * [GSUBO(BC)] ];  !
617    8!                   IF NRSET <> 0 [IFM(BC)] := [MOO] * [GSUBO(BC)] + [MOA];  !
618    8!                ENDIF;                                                  !
619    7!                IF BFLUTR <> 0 OR BGUST <> 0 OR BBLAST <> 0 THEN        !
620    8!                   CALL ROWPART ( [UGTKF], [UGTKO], [UGTKAB], [PFOA(BC)] );  !
621    8!                   [TMP1]  := TRANS( [UGTKO] ) * [GSUBO(BC)];           !
622    8!                   CALL TRNSPOSE ( [TMP1], [TMP2] );                    !
623    8!                   [UGTKA] := [UGTKAB] + [TMP2];                        !
624    8!                ENDIF;                                                  !
625    7!             ELSE                                                    !
626    7!$                                                                    $!
627    7!$               NO F-SET REDUCTION                                   $!
628    7!$                                                                    $!
629    7!                [KAA] := [KFF];                                      !
630    7!                IF BLOAD <> 0 [PA] := [PF];                          !
631    7!                IF BFLUTR <> 0 OR BGUST <> 0 OR BBLAST <> 0 [UGTKA]:=[UGTKF];  !
```

```
632    7!                IF BMASS <> 0 [MAA] := [MFF];                                  !
633    7!             ENDIF;                                                            !
634    6!          ENDIF;                                                               !
635    5!$                                                                            $!
636    5!          IF NRSET <> 0 THEN                                                   !
637    6!$                                                                            $!
638    6!$            PERFORM THE SUPPORT SET REDUCTION                               $!
639    6!$                                                                            $!
640    6!             PRINT("LOG=('          SUPPORT REDUCTION')");                   !
641    6!             IF NITER = 1 THEN                                                !
642    7!                CALL PARTN  ( [KAA], [KRR], [KLR], , [KLL], [PARL(BC)] );    !
643    7!                CALL SDCOMP ( [KLL], [KLLINV(BC)], USET(BC), SINGLSET );     !
644    7!                CALL FBS ( [KLLINV(BC)], [KLR], [D(BC)], -1 );               !
645    7!                CALL RBCHECK ( BC, USET(BC), BGPDT(BC), [D(BC)], [KLL],      !
646    7!                               [KRR], [KLR] );                               !
647    7!             ELSE                                                             !
648    7!                IF BLOAD <> 0 THEN                                           !
649    8!                   CALL PARTN  ( [KAA], , [KLR], , [KLL], [PARL(BC)] );     !
650    8!                   CALL SDCOMP ( [KLL], [KLLINV(BC)], USET(BC), SINGLSET );  !
651    8!                ENDIF;                                                       !
652    7!             ENDIF;                                                           !
653    6!$                                                                            $!
654    6!$            CALCULATE THE REDUCED MASS MATRIX                               $!
655    6!$                                                                            $!
656    6!             CALL PARTN ([MAA], [MRRBAR], [MLR], , [MLL], [PARL(BC)]);       !
657    6!             [IFR(BC)] := [MLL] * [D(BC)] + [MLR];                           !
658    6!             [MRR(BC)] := [MRRBAR] + TRANS ( [MLR] )   * [D(BC)] +           !
659    6!                                     TRANS ( [D(BC)] ) * [IFR(BC)];          !
660    6!             [R22]     := TRANS ( [D(BC)] ) * [MLR] + [MRRBAR];              !
661    6!$                                                                            $!
662    6!             IF BLOAD <> 0 THEN                                              !
663    7!$                                                                            $!
664    7!$               PROCESS STATICS WITH INERTIA RELIEF                          $!
665    7!$                                                                            $!
666    7!                PRINT(                                                       !
667    7!                  "LOG=('          >>>DISCIPLINE: STATICS(INERTIA RELIEF)')"); !
668    7!                CALL ROWPART ( [PA], [PR], [PLBAR], [PARL(BC)] );            !
669    7!                [LHS(BC)] := [MRR(BC)];                                      !
670    7!                [RHS(BC)] := TRANS([D(BC)]) * [PLBAR] + [PR];                !
671    7!                CALL INERTIA ( [LHS(BC)], [RHS(BC)], [AR] );                 !
672    7!                [AL]      := [D(BC)] * [AR];                                 !
673    7!                CALL ROWMERGE ( [AA], [AR], [AL], [PARL(BC)] );              !
674    7!                [RHS(BC)] := [PLBAR] - [IFR(BC)] * [AR];                     !
675    7!                CALL FBS ( [KLLINV(BC)], [RHS(BC)], [UL] );                  !
676    7!                CALL YSMERGE ( [UA], , [UL], [PARL(BC)] );                   !
677    7!             ENDIF;                                                          !
678    6!             IF BMODES <> 0 THEN                                             !
679    7!                PRINT("LOG=('          >>>DISCIPLINE: NORMAL MODES')");      !
680    7!                CALL REIG ( NITER, BC, USET(BC), [KAA], [MAA], [MRR(BC)],    !
681    7!                            [D(BC)], LAMBDA, [PHIA], [MII], HSIZE(BC) );     !
682    7!                CALL OFPMROOT ( NITER, BC, NUMOPTBC, LAMBDA );               !
683    7!                CALL FCEVAL ( NITER, BC, LAMBDA, CONST );                    !
684    7!             ENDIF;                                                          !
685    6!          ELSE                                                               !
686    6!$                                                                            $!
687    6!$            NO SUPPORT SET REDUCTION                                        $!
688    6!$                                                                            $!
689    6!             IF BLOAD <> 0 THEN                                              !
690    7!                PRINT("LOG=('          >>>DISCIPLINE: STATICS')");           !
691    7!                CALL SDCOMP ( [KAA], [KLLINV(BC)], USET(BC), SINGASET  );    !
692    7!                CALL FBS ( [KLLINV(BC)], [PA], [UA] );                       !
693    7!             ENDIF;                                                          !
694    6!             IF BMODES <> 0 THEN                                             !
695    7!                PRINT("LOG=('          >>>DISCIPLINE: NORMAL MODES')");      !
696    7!                CALL REIG ( NITER, BC, USET(BC), [KAA], [MAA], , , LAMBDA,   !
697    7!                            [PHIA], [MII], HSIZE(BC) );                      !
698    7!                CALL OFPMROOT ( NITER, BC, NUMOPTBC, LAMBDA );               !
699    7!                CALL FCEVAL ( NITER, BC, LAMBDA, CONST );                    !
700    7!             ENDIF;                                                          !
701    6!          ENDIF;                                                             !
702    5!       ENDIF;                                                                !
703    4!       IF BSAERO <> 0 THEN                                                   !
704    5!$                                                                            $!
705    5!$         PERFORM STATIC AEROELASTIC ANALYSES                                $!
706    5!$                                                                            $!
707    5!          PRINT("LOG=('          SAERO INITIALIZATION')");                   !
708    5!$****************************** TAKEN OUT FOR ZAERO ***************************$! ←
709    5!$         CALL TRNSPOSE ( [GSTKF], [GSKF] );                                 $!
710    5!$*********************************************************************************$!
711    5!          CALL TRNSPOSE ( [UGTKF], [GSKF] );                                 ! ←
712    5!          LOOP := TRUE;                                                      !
```

```
713    5!            SUB  := 0;                                                          !
714    5!            WHILE LOOP DO                                                      !
715    6!               SUB := SUB + 1;                                                 !
716    6!               CALL SAERODRV (BC, SUB, LOOP, MINDEX, SYM, MACH, QDP, 1 );      !
717    6!$                                                                            $!
718    6!$               ADJUST THE KFF MATRIX AND DETERMINE THE RIGID AIR LOADS       $!
719    6!$                                                                            $!
720    6!$****************************** TAKEN OUT FOR ZAERO ***************************$!  <-
721    6!$               IF SYM = 1  [AICS] := [GTKF]*[TRANS([AICMAT(MINDEX)])*[GSKF]];  $!
722    6!$               IF SYM = -1 [AICS] := [GTKF]*[TRANS([AAICMAT(MINDEX)])*[GSKF]]; $!
723    6!$*******************************************************************************$!
724    6!                IF SYM = 1  [AICS] := [UGTKF]*[TRANS([AICMAT(MINDEX)])*[GSKF]];  !
725    6!                IF SYM = -1 [AICS] := [UGTKF]*[TRANS([AAICMAT(MINDEX)])*[GSKF]]; !
726    6!                [PAF]  := (QDP) [ [UGTKF] * [AIRFRC(MINDEX)] ];                  !  <-
727    6!                [KAFF] := [KFF] - (QDP) [AICS];                                  !
728    6!$                                                                            $!
729    6!$               REDUCE THE MATRICES WITH AEROELASTIC CORRECTIONS              $!
730    6!$               SAVE THE SUBCASE/BC DEPENDENT DATA FOR SENSITIVITY ANALYSIS   $!
731    6!$                                                                            $!
732    6!                IF NGDR <> 0 THEN                                              !
733    7!$                                                                            $!
734    7!$                  PERFORM THE GENERAL DYNAMIC REDUCTION                     $!
735    7!$                                                                            $!
736    7!                   PRINT("LOG=('         SAERO DYNAMIC REDUCTION')");        !
737    7!                   [MAAA] := TRANS ( [GSUBO(BC)] ) * [ [MFF] * [GSUBO(BC)] ];  !
738    7!                   [KAAA] := TRANS ( [GSUBO(BC)] ) * [ [KAFF] * [GSUBO(BC)] ];  !
739    7!                   [PAA]  := TRANS ( [GSUBO(BC)] ) * [PAF];                    !
740    7!                ELSE                                                           !
741    7!                   IF NOMIT <> 0 THEN                                          !
742    8!$                                                                            $!
743    8!$                     PERFORM THE STATIC REDUCTION                           $!
744    8!$                                                                            $!
745    8!                      PRINT("LOG=('         SAERO STATIC CONDENSATION')");   !
746    8!$                                                                            $!
747    8!                      IF NITER = 1  AND SUB = 1 AND NRSET <> 0 AND BLOAD = 0 AND !
748    9!                         BMODES = 0 AND BFLUTR = 0 AND BDYN = 0 THEN         !
749    9!$                                                                            $!
750    9!$                        FORM [KAA] ON FIRST PASS SO [D] CAN BE FORMED       $!
751    9!$                                                                            $!
752    9!                         CALL FREDUCE ([KFF], , [PFOA(BC)], , [KOOINV(BC)], , ,  !
753    9!                                       [GSUBO(BC)], [KAA], , , USET(BC) );   !
754    9!                      ENDIF;                                                  !
755    8!$                                                                            $!
756    8!                      CALL FREDUCE ( [KAFF], [PAF], [PFOA(BC)], BSAERO,       !
757    8!                                     [KOOL(BC,SUB)], [KOOU(BC,SUB)],          !
758    8!                                     [KAO(BC,SUB)], [GASUBO(BC,SUB)], [KAAA],  !
759    8!                                     [PAA], [POARO(BC,SUB)], USET(BC));       !
760    8!$                                                                            $!
761    8!                      IF BMASS <> 0 THEN                                      !
762    9!$                                                                            $!
763    9!$                        PERFORM GUYAN REDUCTION OF THE MASS MATRIX          $!
764    9!$                                                                            $!
765    9!                         CALL PARTN ( [MFF], [MOO], , [MOA], [MAABAR],        !
766    9!                                      [PFOA(BC)] );                           !
767    9!                         [MAAA] := [MAABAR] + TRANS([MOA]) * [GASUBO(BC,SUB)] +  !
768    9!                                   TRANS([GASUBO(BC,SUB)]) * [MOA] +          !
769    9!                                   TRANS([GASUBO(BC,SUB)]) * [[MOO] *         !
770    9!                                   [GASUBO(BC,SUB)]];                         !
771    9!                         IF NRSET <> 0                                        !
772   10!                                [IFMA(BC,SUB)] := [MOO]*[GASUBO(BC,SUB)]+[MOA]; !
773    9!                      ENDIF;                                                  !
774    8!                   ELSE                                                       !
775    8!$                                                                            $!
776    8!$                     NO F-SET REDUCTION                                     $!
777    8!$                                                                            $!
778    8!                      IF NITER = 1  AND SUB = 1 AND NRSET <> 0 AND BLOAD = 0 AND !
779    9!                         BMODES = 0 AND BFLUTR = 0 AND BDYN = 0 THEN         !
780    9!$                                                                            $!
781    9!$                        FORM [KAA] ON FIRST PASS SO [D] CAN BE FORMED       $!
782    9!$                                                                            $!
783    9!                         [KAA] := [KFF];                                      !
784    9!                      ENDIF;                                                  !
785    8!                      [KAAA] := [KAFF];                                       !
786    8!                      [MAAA] := [MFF];                                        !
787    8!                      [PAA]  := [PAF];                                        !
788    8!                   ENDIF;                                                     !
789    7!                ENDIF;                                                        !
790    6!$                                                                            $!
791    6!                IF NRSET <> 0 THEN                                             !
792    7!$                                                                            $!
793    7!$                  PERFORM THE SUPPORT SET REDUCTION                         $!
```

```
794    7!$                                                                    $!
795    7!                 PRINT("LOG=('         SAERO SUPPORT REDUCTION')");   !
796    7!$                                                                    $!
797    7!                 IF NITER = 1  AND SUB  = 1 AND BLOAD = 0 AND BMODES = 0 AND  !
798    8!                    BFLUTR = 0 AND BDYN = 0 THEN                     !
799    8!$                                                                    $!
800    8!$                   [D] WAS NOT COMPUTED FOR NON-SAERO DISCIPLINES SO $!
801    8!$                   NEED TO COMPUTE IT NOW                           $!
802    8!$                                                                    $!
803    8!                    CALL PARTN  ( [KAA], [KRR], [KLR], , [KLL], [PARL(BC)] );  !
804    8!                    CALL SDCOMP ( [KLL], [KLLINV(BC)], USET(BC), SINGLSET );   !
805    8!                    CALL FBS ( [KLLINV(BC)], [KLR], [D(BC)], -1 );   !
806    8!                    CALL RBCHECK ( BC, USET(BC), BGPDT(BC), [D(BC)], [KLL],    !
807    8!                                   [KRR], [KLR] );                   !
808    8!                 ENDIF;                                              !
809    7!$                                                                    $!
810    7!$                 CALCULATE THE REDUCED MASS MATRIX                  $!
811    7!$                                                                    $!
812    7!                 CALL PARTN ([MAAA], [MRRBAR], [MLR], , [MLL], [PARL(BC)]);  !
813    7!                 [R13(BC,SUB)] := [MLL] * [D(BC)] + [MLR];           !
814    7!                 [R33]         := [MRRBAR] + TRANS ( [MLR] ) * [D(BC)] +    !
815    7!                                  TRANS ( [D(BC)] ) * [R13(BC,SUB)];   !
816    7!                 [R22]         := TRANS ( [D(BC)] ) * [MLR] + [MRRBAR];     !
817    7!                 CALL TRNSPOSE ( [R13(BC,SUB)], [R21(BC,SUB)] );      !
818    7!$                                                                    $!
819    7!$                 PROCESS STEADY AEROELASTIC DISCIPLINE              $!
820    7!$                                                                    $!
821    7!                 PRINT("LOG=('         >>>DISCIPLINE: STEADY AERO')");  !
822    7!                 CALL PARTN ( [KAAA], [KARR], [R12(BC,SUB)], [KARL], [R11],   !
823    7!                              [PARL(BC)] );                          !
824    7!                 [R32(BC,SUB)] := TRANS([D(BC)]) * [R12(BC,SUB)] + [KARR];    !
825    7!                 [R31(BC,SUB)] := TRANS([D(BC)]) * [R11] + [KARL];    !
826    7!$                                                                    $!
827    7!                 CALL DECOMP ( [R11], [RL11(BC,SUB)], [RU11(BC,SUB)] );  !
828    7!$                                                                    $!
829    7!                 CALL ROWPART ( [PAA], [PARBAR], [PAL], [PARL(BC)] );  !
830    7!                 CALL GFBS ( [RL11(BC,SUB)], [RU11(BC,SUB)], [PAL],    !
831    7!                             [R11PAL(BC,SUB)], -1);                   !
832    7!                 [PRIGID] := [PARBAR] + TRANS([D(BC)]) * [PAL];       !
833    7!                 [P1]     := [R21(BC,SUB)] * [R11PAL(BC,SUB)];        !
834    7!                 [P2]     := [PRIGID] + [R31(BC,SUB)] * [R11PAL(BC,SUB)];  !
835    7!$                                                                    $!
836    7!                 CALL GFBS ( [RL11(BC,SUB)], [RU11(BC,SUB)], [R12(BC,SUB)],   !
837    7!                             [R1112(BC,SUB)], -1);                    !
838    7!                 CALL GFBS ( [RL11(BC,SUB)], [RU11(BC,SUB)], [R13(BC,SUB)],   !
839    7!                             [R1113(BC,SUB)], -1);                    !
840    7!                 [K11]         := [R22] + [R21(BC,SUB)] * [R1112(BC,SUB)];  !
841    7!                 [K12(BC,SUB)] := [R21(BC,SUB)] * [R1113(BC,SUB)];    !
842    7!                 [K21(BC,SUB)] := [R32(BC,SUB)] +                     !
843    7!                                  [R31(BC,SUB)] * [R1112(BC,SUB)];    !
844    7!                 [K22]         := [R33] + [R31(BC,SUB)] * [R1113(BC,SUB)];  !
845    7!$                                                                    $!
846    7!                 CALL DECOMP ( [K11], [KL11(BC,SUB)], [KU11(BC,SUB)] );   !
847    7!                 CALL GFBS ( [KL11(BC,SUB)], [KU11(BC,SUB)], [P1],     !
848    7!                             [PAR(BC,SUB)] );                         !
849    7!                 CALL GFBS ( [KL11(BC,SUB)], [KU11(BC,SUB)], [K12(BC,SUB)],   !
850    7!                             [K1112(BC,SUB)],-1);                     !
851    7!                 [LHSA(BC,SUB)] := [K22] + [K21(BC,SUB)] * [K1112(BC,SUB)];   !
852    7!                 [RHSA(BC,SUB)] := [P2] - [K21(BC,SUB)] * [PAR(BC,SUB)];      !
853    7!$****************** CALL SAERO NOW ! ***************************************$!
854    7!                 CALL SAERO ( NITER, BC, MINDEX, SUB, SYM, QDP, STABCF,    !
855    7!                              BGPDT(BC), [LHSA(BC,SUB)], [RHSA(BC,SUB)], [AAR],!
856    7!                              [DELTA(SUB)], [PRIGID], [R33],           !
857    7!                              CONST, AEFLG(SUB), [AARC], [DELC]);      !
858    7!$*************************************************************************$!
859    7!                 [AAL] := [D(BC)] * [AAR];                           !
860    7!                 CALL ROWMERGE ( [AAA(SUB)], [AAR], [AAL], [PARL(BC)] );   !
861    7!                 [UAR] := [K1112(BC,SUB)] * [AAR] + [PAR(BC,SUB)] *   !
862    7!                          [DELTA(SUB)];                               !
863    7!                 [UAL] := [R1112(BC,SUB)] * [UAR] + [R1113(BC,SUB)] * [AAR]    !
864    7!                          - [R11PAL(BC,SUB)] * [DELTA(SUB)];          !
865    7!                 CALL ROWMERGE ( [UAA(SUB)], [UAR], [UAL], [PARL(BC)] );   !
866    7!                 IF NOMIT <> 0 [PAO(SUB)] := [POARO(BC,SUB)] * [DELTA(SUB)] ;  !
867    7!                 IF AEFLG(SUB) THEN                                  !
868    8!                    [AAL]         := [D(BC)] * [AARC];               !
869    8!                    CALL ROWMERGE ( [AAAC(SUB)], [AARC], [AAL], [PARL(BC)] );  !
870    8!                    [UAR]    := [K1112(BC,SUB)] * [AARC] + [PAR(BC,SUB)] *    !
871    8!                                [DELC];                               !
872    8!                    [UAL]    := [R1112(BC,SUB)] * [UAR] +              !
873    8!                                [R1113(BC,SUB)] * [AARC] -            !
874    8!                                [R11PAL(BC,SUB)] * [DELC];            !
```

```
8!                 CALL ROWMERGE ( [UAAC(SUB)], [UAR], [UAL], [PARL(BC)] );
8!                 IF NOMIT <> 0 [PAOC(SUB)] := [POARO(BC,SUB)]*[DELC] ;
8!              ENDIF;
7!           ELSE
7!$
7!$             NO SUPPORT SET REDUCTION
7!$             PROCESS STEADY AEROELASTIC DISCIPLINE
7!$
7!              PRINT("LOG=('         >>>DISCIPLINE: STEADY AERO')");
7!$
7!           ENDIF;
6!        ENDDO;
5!     ENDIF;
4!$
4!$    PERFORM ANY DYNAMIC ANALYSES -- NOTE THAT THESE ARE INDEPENDENT
4!$    OF THE SUPPORT SET
4!$
4!     IF BDYN <> 0 THEN
5!        IF BFLUTR <> 0 THEN
6!           PRINT("LOG=('         >>>DISCIPLINE: FLUTTER')");
6!           SUB  := 0;
6!           LOOP := TRUE;
6!           WHILE LOOP DO
7!              SUB := SUB + 1;
7!              CALL FLUTDRV ( BC, SUB, LOOP );
7!              CALL FLUTQHHZ ( NITER, BC, SUB, ESIZE(BC), PSIZE(BC), [AJK],
7!                             [SKJ],[UGTKA], [PHIA], USET(BC),
7!                             [TMN(BC)], [GSUBO(BC)], NGDR, AECOMPZ, GEOMZA,
7!                             [PHIKH], [QHHLFL(BC,SUB)], OAGRDDSP );
7!              CALL FLUTDMA ( NITER, BC, SUB, ESIZE(BC), PSIZE(BC),
7!                        BGPDT(BC), USET(BC), [MAA], [KAA], [TMN(BC)],
7!                        [GSUBO(BC)], NGDR, LAMBDA, [PHIA],
7!                        [MHHFL(BC,SUB)], [BHHFL(BC,SUB)], [KHHFL(BC,SUB)] );
7!              CALL FLUTTRAZ ( NITER, BC, SUB, [QHHLFL(BC,SUB)], LAMBDA,
7!                             HSIZE(BC), ESIZE(BC), [MHHFL(BC,SUB)],
7!                             [BHHFL(BC,SUB)], [KHHFL(BC,SUB)],
7!                             CLAMBDA, CONST,AEROZ );
7!           ENDDO;
6!        ENDIF;
5!$
5!        IF BDRSP <> 0 THEN
6!           IF BMTR   <> 0 OR BDTR <> 0 THEN
7!              PRINT("LOG=('         >>>DISCIPLINE: TRANSIENT RESPONSE')");
7!           ENDIF;
6!           IF BMFR <> 0 OR BDFR <> 0 THEN
7!              PRINT("LOG=('         >>>DISCIPLINE: FREQUENCY RESPONSE')");
7!           ENDIF;
6!$$$$$$$$$$$$$$$$$$$$$$ MODIFIED FOR ZAERO $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
6!$           CALL QHHLGEN (BC, ESIZE(BC), [QKKL], [QKJL], [UGTKA], [PHIA],
6!$                         [PHIKH], [QHHL], [QHJL]);
6!$$$$$$$$$$$$$$$$$$$$$$ MODIFIED FOR ZAERO $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
6!           CALL QHHLGENZ (BC, ESIZE(BC),[AJK],[SKJ],[QGK],[UGTKA], [PHIA],
6!                         [PHIKH], [QHHL], [QHJL], AEROZ);
6!           CALL DMA ( NITER, BC, ESIZE(BC), PSIZE(BC), BGPDT(BC), USET(BC),
6!                      [MAA], [KAA], [TMN(BC)], [GSUBO(BC)], NGDR,
6!                      LAMBDA, [PHIA], [MDD], [BDD], [KDDT], [KDDF],
6!                      [MHH], [BHH], [KHHT], [KHHF] );
6!           CALL DYNLOAD ( NITER, BC, GSIZE, ESIZE(BC), PSIZE(BC), SMPLOD,
6!                          BGPDT(BC), USET(BC), [TMN(BC)], [GSUBO(BC)],
6!                          NGDR, [PHIA], [QHJL], [PDT], [PDF],
6!                          [PTGLOAD], [PTHLOAD], [PFGLOAD], [PFHLOAD] );
6!           CALL DYNRSP (BC, ESIZE(BC), [MDD], [BDD], [KDDT], [KDDF],
6!                        [MHH], [BHH], [KHHT], [KHHF], [PDT], [PDF],
6!                        [QHHL], [UTRANA], [UFREQA], [UTRANI], [UFREQI],
6!                        [UTRANE], [UFREQE] );
6!           IF BMTR <> 0 [UTRANA] := [PHIA] * [UTRANI];
6!           IF BMFR <> 0 [UFREQA] := [PHIA] * [UFREQI];
6!        ENDIF;
5!     ENDIF;
4!     IF BBLAST <> 0 THEN
5!        PRINT("LOG=('         >>>DISCIPLINE: BLAST')");
5!        CALL BLASTFIT ( BC, [QJJL], [MATTR], [MATSS], BQDP, [BFRC],
5!                        [DWNWSH],  HSIZE(BC), [ID2], [MPART], [UGTKA],
5!                        [BLGTJA], [BLSTJA] );
5!        CALL COLPART ( [PHIA], , [PHIE], [MPART] );
5!        CALL ROWMERGE ( [PHIR], [ID2], [D(BC)], [PARL(BC)] );
5!        CALL COLMERGE ( [PHIB], [PHIR], [PHIE], [MPART] );
5!        [GENM]    := TRANS( [PHIB] ) * [ [MAA] * [PHIB] ];
5!        [GENK]    := TRANS( [PHIB] ) * [ [KAA] * [PHIB] ];
5!        [DTSLP]   := TRANS ( [BLSTJA] ) * [PHIB];
5!        [FTF]     := TRANS ( [PHIB] ) * [BLGTJA];
```

```
956    5!            [GENF]    := (BQDP) [FTF] * [BFRC];                                          !
957    5!            [GENFA]   := (BQDP) [FTF] * [MATSS];                                         !
958    5!            [GENQ]    := [GENFA] * [DTSLP];                                              !
959    5!            [GENQL]   := (BQDP) [FTF] * [MATTR];                                         !
960    5!            CALL PARTN ( [GENQ],[QRR] , , [QRE], [QEE], [MPART] );                       !
961    5!            CALL PARTN ( [GENK], , , , [KEE], [MPART] );                                 !
962    5!            [KEQE]    := [QEE] + [KEE];                                                  !
963    5!            CALL DECOMP ( [KEQE], [LKQ], [UKQ] );                                       !
964    5!            CALL ROWPART ( [GENF], [GFR], [GFE], [MPART] );                              !
965    5!            CALL GFBS ( [LKQ], [UKQ], [GFE], [BTEM] );                                  !
966    5!            [DELM]    := -[QRE] * [BTEM] + [GFR];                                        !
967    5!            CALL BLASTRIM ( BC, [DELM], [MRR(BC)], [URDB], [DELB] );                     !
968    5!            [ELAS]    := [BTEM] * [DELB];                                                !
969    5!            [SLPMOD] := TRANS ( [BLSTJA] ) * [PHIE];                                     !
970    5!            CALL BLASTDRV ( BC, [GENM], [GENK], [GENFA], [GENQL], [DELB],                !
971    5!                            [URDB], [DWNWSH], [SLPMOD], [ELAS], [UBLASTI] );             !
972    5!         ENDIF;                                                                          !
973.   4!$                                                                                        $!
974    4!$        BEGIN THE DATA RECOVERY OPERATIONS                                              $!
975    4!$                                                                                        $!
976    4!         PRINT("LOG=('          DATA RECOVERY')");                                       !
977    4!         IF NUMOPTBC > 1 CALL NULLMAT ([UF], [AF], [PHIF], [UTRANF], [UFREQF]);          !
978    4!         IF NGDR <> 0 THEN                                                               !
979    5!$                                                                                        $!
980    5!$           DATA RECOVERY WITH GDR                                                       $!
981    5!$           APPEND THE GDR-GENERATED DOFS TO THE F-SET                                   $!
982    5!$                                                                                        $!
983    5!            PRINT("LOG=('          DYNAMIC REDUCTION RECOVERY')");                       !
984    5!            IF BLOAD <> 0 THEN                                                           !
985    6!               [UFGDR] := [GSUBO(BC)] * [UA];                                            !
986    6!               CALL ROWPART ( [UA], [UJK], , [PAJK] );                                   !
987    6!               CALL ROWMERGE ( [UF], [UJK], [UFGDR], [PFJK] );                           !
988    6!               IF NRSET <> 0 THEN                                                        !
989    7!                  [AFGDR] := [GSUBO(BC)] * [AA];                                         !
990    7!                  CALL ROWPART ( [AA], [UJK], , [PAJK] );                                !
991    7!                  CALL ROWMERGE ( [AF], [UJK], [AFGDR], [PFJK] );                        !
992    7!               ENDIF;                                                                    !
993    6!            ENDIF;                                                                       !
994    5!            IF BSAERO <> 0 THEN                                                          !
995    6!               FOR S = 1 TO SUB DO                                                       !
996    7!                  [UFGDR] := [GSUBO(BC)] * [UAA(S)];                                     !
997    7!                  CALL ROWPART ( [UAA(S)], [UJK], , [PAJK] );                            !
998    7!                  CALL ROWMERGE ( [UAFTMP], [UJK], [UFGDR], [PFJK] );                    !
999    7!$                                                                                        $!
1000   7!$                 MERGE THE CURRENT SUBCASE DEPENDENT RESULTS INTO A SINGLE              $!
1001   7!$                 MATRIX OF RESPONSE QUANTITIES FOR FURTHER RECOVERY                     $!
1002   7!$                                                                                        $!
1003   7!                  CALL SAEROMRG ( BC, S, [UAF], [UAFTMP] );                              !
1004   7!                  IF NRSET <> 0 THEN                                                     !
1005   8!                     [AFGDR] := [GSUBO(BC)] * [AAA(S)];                                  !
1006   8!                     CALL ROWPART ( [AAA(S)], [UJK], , [PAJK] );                         !
1007   8!                     CALL ROWMERGE ( [AAFTMP], [UJK], [AFGDR], [PFJK] );                 !
1008   8!                     CALL SAEROMRG ( BC, S, [AAF], [AAFTMP] );                           !
1009   8!                  ENDIF;                                                                 !
1010   7!                  IF AEFLG(S) THEN                                                       !
1011   8!                     [UFGDR] := [GSUBO(BC)] * [UAAC(S)];                                 !
1012   8!                     CALL ROWPART ( [UAAC(S)], [UJK], , [PAJK] );                        !
1013   8!                     CALL ROWMERGE ( [UAFC(S)], [UJK], [UFGDR], [PFJK] );               !
1014   8!                     [AFGDR] := [GSUBO(BC)] * [AAAC(S)];                                 !
1015   8!                     CALL ROWPART ( [AAAC(S)], [UJK], , [PAJK] );                        !
1016   8!                     CALL ROWMERGE ( [AAFC(S)], [UJK], [AFGDR], [PFJK] );               !
1017   8!                  ENDIF;                                                                 !
1018   7!               ENDDO;                                                                    !
1019   6!            ENDIF;                                                                       !
1020   5!            IF BMODES <> 0 THEN                                                          !
1021   6!               [UFGDR]   := [GSUBO(BC)] * [PHIA];                                        !
1022   6!               CALL ROWPART ( [PHIA], [UJK], , [PAJK] );                                 !
1023   6!               CALL ROWMERGE ( [PHIF], [UJK], [UFGDR], [PFJK] );                         !
1024   6!            ENDIF;                                                                       !
1025   5!            IF BDTR   <> 0 OR BMTR <> 0 THEN                                             !
1026   6!               [UFGDR] := [GSUBO(BC)] * [UTRANA];                                        !
1027   6!               CALL ROWPART ( [UTRANA], [UJK], , [PAJK] );                               !
1028   6!               CALL ROWMERGE ( [UTRANF], [UJK], [UFGDR], [PFJK] );                       !
1029   6!            ENDIF;                                                                       !
1030   5!            IF BDFR   <> 0 OR BMFR <> 0 THEN                                             !
1031   6!               [UFGDR] := [GSUBO(BC)] * [UFREQA];                                        !
1032   6!               CALL ROWPART ( [UFREQA], [UJK], , [PAJK] );                               !
1033   6!               CALL ROWMERGE ( [UFREQF], [UJK], [UFGDR], [PFJK] );                       !
1034   6!            ENDIF;                                                                       !
1035   5!         ELSE                                                                            !
1036   5!            IF NOMIT <> 0 THEN                                                           !
```

```
1037    6!$                                                                  $!
1038    6!$           DATA RECOVERY WITH STATIC CONDENSATION                 $!
1039    6!$                                                                  $!
1040    6!            PRINT("LOG=('         STATIC CONDENSATION RECOVERY')");  !
1041    6!            IF BLOAD <> 0 THEN                                       !
1042    7!               CALL RECOVA ( [UA], [PO], [GSUBO(BC)], NRSET, [AA],   !
1043    7!                             [IFM(BC)], , [KOOINV(BC)],,[PFOA(BC)], [UF] );  !
1044    7!               IF NRSET <> 0 CALL RECOVA ( [AA], , [GSUBO(BC)],,,,,,,  !
1045    8!                                           [PFOA(BC)], [AF] );        !
1046    7!            ENDIF;                                                   !
1047    6!            IF BSAERO <> 0 THEN                                      !
1048    7!               FOR S = 1 TO SUB DO                                   !
1049    8!                  CALL RECOVA ( [UAA(S)], [PAO(S)], [GASUBO(BC,S)],  !
1050    8!                                NRSET, [AAA(S)], [IFMA(BC,S)], BSAERO,  !
1051    8!                                [KOOL(BC,S)], [KOOU(BC,S)],          !
1052    8!                                [PFOA(BC)], [UAFTMP] );              !
1053    8!$                                                                   $!
1054    8!$               MERGE THE CURRENT SUBCASE DEPENDENT RESULTS INTO A SINGLE $!
1055    8!$               MATRIX OF RESPONSE QUANTITIES FOR FURTHER RECOVERY  $!
1056    8!$                                                                  $!
1057    8!                CALL SAEROMRG ( BC, S, [UAF], [UAFTMP] );            !
1058    8!                IF NRSET <> 0 THEN                                   !
1059    9!                   CALL RECOVA ( [AAA(S)],,[GASUBO(BC,S)],,,,,,,     !
1060    9!                                 [PFOA(BC)], [AAFTMP]);              !
1061    9!                   CALL SAEROMRG ( BC, S, [AAF], [AAFTMP] );         !
1062    9!                ENDIF;                                               !
1063    8!                IF AEFLG(S) THEN                                     !
1064    9!                   CALL RECOVA ( [UAAC(S)], [PAOC(S)], [GASUBO(BC,S)],  !
1065    9!                                 NRSET, [AAAC(S)], [IFMA(BC,S)], BSAERO,  !
1066    9!                                 [KOOL(BC,S)], [KOOU(BC,S)],         !
1067    9!                                 [PFOA(BC)], [UAFC(S)] );            !
1068    9!                   CALL RECOVA ( [AAAC(S)],,[GASUBO(BC,S)],,,,,,,    !
1069    9!                                 [PFOA(BC)], [AAFC(S)]);             !
1070    9!                ENDIF;                                               !
1071    8!             ENDDO;                                                  !
1072    7!            ENDIF;                                                   !
1073    6!            IF BMODES <> 0 THEN                                      !
1074    7!               [PHIO] := [GSUBO(BC)] * [PHIA];                       !
1075    7!               CALL ROWMERGE ( [PHIF], [PHIO], [PHIA], [PFOA(BC)] );  !
1076    7!            ENDIF;                                                   !
1077    6!            IF BDTR <> 0 OR BMTR <> 0 THEN                           !
1078    7!               CALL RECOVA ( [UTRANA], , [GSUBO(BC)],,,,,,,          !
1079    7!                                         [PFOA(BC)], [UTRANF] );     !
1080    7!            ENDIF;                                                   !
1081    6!            IF BDFR <> 0 OR BMFR <> 0 THEN                           !
1082    7!               CALL RECOVA ( [UFREQA], , [GSUBO(BC)],,,,,,,          !
1083    7!                                         [PFOA(BC)], [UFREQF] );     !
1084    7!            ENDIF;                                                   !
1085    6!         ELSE                                                        !
1086    6!$                                                                  $!
1087    6!$            DATA RECOVERY WITHOUT F-SET REDUCTION                  $!
1088    6!$                                                                  $!
1089    6!            IF BLOAD <> 0 THEN                                       !
1090    7!               [UF] := [UA];                                         !
1091    7!               IF NRSET <> 0 [AF]    := [AA];                        !
1092    7!            ENDIF;                                                   !
1093    6!            IF BSAERO <> 0 THEN                                      !
1094    7!               FOR S = 1 TO SUB DO                                   !
1095    8!$                                                                  $!
1096    8!$               MERGE THE CURRENT SUBCASE DEPENDENT RESULTS INTO A SINGLE $!
1097    8!$               MATRIX OF RESPONSE QUANTITIES FOR FURTHER RECOVERY  $!
1098    8!$                                                                  $!
1099    8!                CALL SAEROMRG ( BC, S, [UAF], [UAA(S)] );            !
1100    8!                IF NRSET <> 0 CALL SAEROMRG ( BC, S, [AAF], [AAA(S)] );  !
1101    8!                IF AEFLG(S) THEN                                     !
1102    9!                   [UAFC(S)] := [UAAC(S)];                           !
1103    9!                   [AAFC(S)] := [AAAC(S)];                           !
1104    9!                ENDIF;                                               !
1105    8!             ENDDO;                                                  !
1106    7!            ENDIF;                                                   !
1107    6!            IF BMODES <> 0 [PHIF]    := [PHIA];                      !
1108    6!            IF BDTR   <> 0 OR BMTR <> 0 [UTRANF] := [UTRANA];        !
1109    6!            IF BDFR   <> 0 OR BMFR <> 0 [UFREQF] := [UFREQA];        !
1110    6!         ENDIF;                                                      !
1111    5!      ENDIF;                                                         !
1112    4!$                                                                  $!
1113    4!      IF NUMOPTBC > 1 CALL NULLMAT ( [UN], [AN], [PHIN] );          !
1114    4!      IF NSPC <> 0 THEN                                              !
1115    5!$                                                                  $!
1116    5!$         DATA RECOVERY WITH SPC-REDUCTION                         $!
1117    5!$                                                                  $!
```

```
1118    5!            PRINT("LOG=('          SPC RECOVERY')");                          !
1119    5!            IF BLOAD <> 0  THEN                                               !
1120    6!               CALL YSMERGE ( [UN], [YS(BC)], [UF], [PNSF(BC)] );             !
1121    6!               CALL OFPSPCF ( NITER, BC, 1, 1, GSIZE, ESIZE(BC), NGDR,        !
1122    6!                              [KFS], [KSS], [UF], [YS(BC)], [PS],             !
1123    6!                              [PNSF(BC)], [PGMN(BC)], [PFJK], , , ,           !
1124    6!                              BGPDT(BC), OGRIDLOD );                          !
1125    6!               IF NRSET <> 0 CALL YSMERGE ( [AN], , [AF], [PNSF(BC)] );       !
1126    6!            ENDIF;                                                            !
1127    5!            IF BSAERO <> 0 THEN                                               !
1128    6!               CALL YSMERGE ( [UAN], [YS(BC)], [UAF], [PNSF(BC)] );           !
1129    6!               IF NRSET <> 0 CALL YSMERGE ( [AAN], , [AAF], [PNSF(BC)] );     !
1130    6!               FOR S = 1 TO SUB DO                                            !
1131    7!                  IF AEFLG(S) THEN                                            !
1132    8!                     CALL YSMERGE ([UANC(S)], [YS(BC)], [UAFC(S)], [PNSF(BC)]); !
1133    8!                     CALL YSMERGE ([AANC(S)], , [AAFC(S)], [PNSF(BC)]);       !
1134    8!                  ENDIF;                                                      !
1135    7!               ENDDO;                                                         !
1136    6!            ENDIF;                                                            !
1137    5!            IF BMODES <> 0 THEN                                               !
1138    6!               CALL YSMERGE ( [PHIN], [YS(BC)], [PHIF],                       !
1139    6!                                          [PNSF(BC)] );                       !
1140    6!               IF DMODES <> 0   CALL OFPSPCF ( NITER, BC, 2, 1, GSIZE,        !
1141    7!                                               ESIZE(BC), NGDR,              !
1142    7!                                               [KFS], , [PHIF], , ,          !
1143    7!                                               [PNSF(BC)], [PGMN(BC)], [PFJK], !
1144    7!                                                , , , BGPDT(BC), OGRIDLOD );  !
1145    6!            ENDIF;                                                            !
1146    5!            IF BDTR   <> 0 OR BMTR <> 0                                       !
1147    6!                           CALL YSMERGE ( [UTRANN], [YS(BC)], [UTRANF],       !
1148    6!                                          [PNSF(BC)], BDTR );                 !
1149    5!            IF BDFR   <> 0 OR BMFR <> 0                                       !
1150    6!                           CALL YSMERGE ( [UFREQN], [YS(BC)], [UFREQF],       !
1151    6!                                          [PNSF(BC)], BDFR );                 !
1152    5!            IF BBLAST <> 0  THEN                                              !
1153    6!               [UBLASTF] := [PHIF]*[UBLASTI];                                 !
1154    6!               CALL OFPSPCF ( NITER, BC, 8, 1, GSIZE, ESIZE(BC), NGDR,        !
1155    6!                              [KFS], , [UBLASTF], , , [PNSF(BC)], [PGMN(BC)], !
1156    6!                              [PFJK], , , , BGPDT(BC), OGRIDLOD );            !
1157    6!            ENDIF;                                                            !
1158    5!         ELSE                                                                 !
1159    5!$                                                                            $!
1160    5!$            DATA RECOVERY WITHOUT SPC-REDUCTION                             $!
1161    5!$                                                                            $!
1162    5!            IF BLOAD <> 0 THEN                                                !
1163    6!               [UN] := [UF];                                                  !
1164    6!               IF NRSET <> 0 [AN] := [AF];                                    !
1165    6!            ENDIF;                                                            !
1166    5!            IF BSAERO <> 0 THEN                                               !
1167    6!               [UAN] := [UAF];                                                !
1168    6!               IF NRSET <> 0 [AAN] := [AAF];                                  !
1169    6!               FOR S = 1 TO SUB DO                                            !
1170    7!                  IF AEFLG(S) THEN                                            !
1171    8!                     [UANC(S)] := [UAFC(S)];                                  !
1172    8!                     [AANC(S)] := [AAFC(S)];                                  !
1173    8!                  ENDIF;                                                      !
1174    7!               ENDDO;                                                         !
1175    6!            ENDIF;                                                            !
1176    5!            IF BMODES <> 0 [PHIN]   := [PHIF];                              !
1177    5!            IF BDTR   <> 0 OR BMTR <> 0 [UTRANN] := [UTRANA];               !
1178    5!            IF BDFR   <> 0 OR BMFR <> 0 [UFREQN] := [UFREQA];               !
1179    5!         ENDIF;                                                              !
1180    4!$                                                                           $!
1181    4!         IF NUMOPTBC > 1 CALL NULLMAT ( [UG(BC)], [AG(BC)], [UAG(BC)],       !
1182    5!                                        [AAG(BC)], [PHIG(BC)] );            !
1183    4!$                                                                           $!
1184    4!         IF NMPC <> 0 THEN                                                   !
1185    5!$                                                                           $!
1186    5!$            DATA RECOVERY WITH MPC-REDUCTION                               $!
1187    5!$                                                                           $!
1188    5!            PRINT("LOG=('          MPC RECOVERY')");                        !
1189    5!            IF BLOAD <> 0 THEN                                              !
1190    6!               [UM] := [TMN(BC)] * [UN];                                    !
1191    6!               CALL ROWMERGE ( [UG(BC)], [UM], [UN], [PGMN(BC)] );          !
1192    6!               IF NRSET <> 0 THEN                                           !
1193    7!                  [UM] := [TMN(BC)] * [AN];                                 !
1194    7!                  CALL ROWMERGE ( [AG(BC)], [UM], [AN], [PGMN(BC)] );       !
1195    7!               ENDIF;                                                       !
1196    6!            ENDIF;                                                          !
1197    5!            IF BSAERO <> 0 THEN                                             !
1198    6!               [UM] := [TMN(BC)] * [UAN];                                   !
```

```
1199    6!              CALL ROWMERGE ( [UAG(BC)], [UM], [UAN], [PGMN(BC)] );                !
1200    6!              IF NRSET <> 0 THEN                                                   !
1201    7!                 [UM] := [TMN(BC)] * [AAN];                                        !
1202    7!                 CALL ROWMERGE ( [AAG(BC)], [UM], [AAN], [PGMN(BC)] );            !
1203    7!              ENDIF;                                                               !
1204    6!              FOR S = 1 TO SUB DO                                                  !
1205    7!                 IF AEFLG(S) THEN                                                  !
1206    8!                    [UM] := [TMN(BC)] * [UANC(S)];                                 !
1207    8!                    CALL ROWMERGE ([UAGC(BC,S)], [UM], [UANC(S)], [PGMN(BC)]);     !
1208    8!                    [UM] := [TMN(BC)] * [AANC(S)];                                 !
1209    8!                    CALL ROWMERGE ([AAGC(BC,S)], [UM], [AANC(S)], [PGMN(BC)]);     !
1210    8!                 ENDIF;                                                            !
1211    7!              ENDDO;                                                               !
1212    6!           ENDIF;                                                                  !
1213    5!           IF BMODES <> 0 THEN                                                     !
1214    6!              [UM] := [TMN(BC)] * [PHIN];                                          !
1215    6!              CALL ROWMERGE ( [PHIG(BC)], [UM], [PHIN], [PGMN(BC)] );             !
1216    6!           ENDIF;                                                                  !
1217    5!           IF BDTR <> 0 OR BMTR <> 0 THEN                                          !
1218    6!              [UM] := [TMN(BC)] * [UTRANN];                                        !
1219    6!              CALL ROWMERGE ( [UTRANG], [UM], [UTRANN], [PGMN(BC)] );              !
1220    6!           ENDIF;                                                                  !
1221    5!           IF BDFR <> 0 OR BMFR <> 0 THEN                                          !
1222    6!              [UM] := [TMN(BC)] * [UFREQN];                                        !
1223    6!              CALL ROWMERGE ( [UFREQG], [UM], [UFREQN], [PGMN(BC)] );              !
1224    6!           ENDIF;                                                                  !
1225    5!        ELSE                                                                       !
1226    5!$                                                                                 $!
1227    5!$          DATA RECOVERY WITHOUT MPC-REDUCTION                                    $!
1228    5!$                                                                                 $!
1229    5!           IF BLOAD <> 0 THEN                                                      !
1230    6!              [UG(BC)] := [UN];                                                    !
1231    6!              IF NRSET <> 0 [AG(BC)] := [AN];                                      !
1232    6!           ENDIF;                                                                  !
1233    5!           IF BSAERO <> 0 THEN                                                     !
1234    6!              [UAG(BC)] := [UAN];                                                  !
1235    6!              IF NRSET <> 0 [AAG(BC)] := [AAN];                                    !
1236    6!              FOR S = 1 TO SUB DO                                                  !
1237    7!                 IF AEFLG(S) THEN                                                  !
1238    8!                    [UAGC(BC,S)] := [UANC(S)];                                     !
1239    8!                    [AAGC(BC,S)] := [AANC(S)];                                     !
1240    8!                 ENDIF;                                                            !
1241    7!              ENDDO;                                                               !
1242    6!           ENDIF;                                                                  !
1243    5!           IF BMODES <> 0 [PHIG(BC)] := [PHIN];                                    !
1244    5!           IF BDTR   <> 0 OR BMTR <> 0 [UTRANG] := [UTRANN];                       !
1245    5!           IF BDFR   <> 0 OR BMFR <> 0 [UFREQG] := [UFREQN];                       !
1246    5!        ENDIF;                                                                     !
1247    4!$                                                                                 $!
1248    4!$       RECOVER PHYSICAL BLAST DISCIPLINE DISPLACEMENTS                           $!
1249    4!$                                                                                 $!
1250    4!        IF BBLAST <> 0 [UBLASTG] := [PHIG(BC)] * [UBLASTI];                        !
1251    4!$                                                                                 $!
1252    4!$       PERFORM CONSTRAINT EVALUATION FOR STATIC DISCIPLINES                      $!
1253    4!$                                                                                 $!
1254    4!        PRINT("LOG=('          CONSTRAINT EVALUATION')");                          !
1255    4!        IF BLOAD <> 0 THEN                                                         !
1256    5!           CALL DCEVAL ( NITER, BC, [UG(BC)], CONST );                             !
1257    5!           CALL SCEVAL ( NITER, BC, [UG(BC)], [SMAT], TREF, [GLBSIG], CONST );     !
1258    5!        ENDIF;                                                                     !
1259    4!        IF BSAERO <> 0 THEN                                                        !
1260    5!           CALL DCEVAL ( NITER, BC, [UAG(BC)], CONST, BSAERO );                    !
1261    5!           CALL SCEVAL ( NITER, BC, [UAG(BC)], [SMAT], TREF, [GLBSIG], CONST,      !
1262    5!                         BSAERO );                                                 !
1263    5!        ENDIF;                                                                     !
1264    4!$                                                                                 $!
1265    4!$       HANDLE OUTPUT REQUESTS                                                    $!
1266    4!$                                                                                 $!
1267    4!        PRINT("LOG=('          OUTPUT PROCESSING')");                              !
1268    4!        IF BSAERO <> 0 THEN                                                        !
1269    5!$                                                                                 $!
1270    5!$          RECOVER STATIC AEROELASTIC LOADS DATA                                  $!
1271    5!$                                                                                 $!
1272    5!           LOOP := TRUE;                                                           !
1273    5!           SUB  := 0;                                                              !
1274    5!           WHILE LOOP DO                                                           !
1275    6!              SUB := SUB + 1;                                                      !
1276    6!              CALL SAERODRV (BC, SUB, LOOP, MINDEX, SYM, MACH, QDP );              !
1277    6!$                                                                                 $!
1278    6!$             CALL THE TRIMMED LOADS COMPUTATION WITH PROPER MATRICES             $!
1279    6!$                                                                                 $!
```

```
1280    6!              IF SYM = 1  THEN                                                !
1281    7!$********************************* TAKEN OUT FOR ZAERO ***********************$! <-
1282    7!$                CALL OFPALOAD ( NITER, BC, MINDEX, SUB, GSIZE, BGPDT(BC),    $!
1283    7!$                                [GTKG], [GSTKG], QDP, [AIRFRC(MINDEX)],      $!
1284    7!$                                [DELTA(SUB)], [AICMAT(MINDEX)],               $!
1285    7!$                                [UAG(BC)], [MGG], [AAG(BC)], [KFS],           $!
1286    7!$                                [KSS], [UAF], [YS(BC)], [PNSF(BC)],           $!
1287    7!$                                [PGMN(BC)], [PFJK], NGDR, USET(BC),           $!
1288    7!$                                OGRIDLOD );                                   $!
1289    7!$*************************************************************************** $! <-
1290    7!                 CALL OFPALOAD ( NITER, BC, MINDEX, SUB, GSIZE, BGPDT(BC),     !
1291    7!                                 [UGTKG], [UGTKG], QDP, [AIRFRC(MINDEX)],      ! <-
1292    7!                                 [DELTA(SUB)], [AICMAT(MINDEX)],               !
1293    7!                                 [UAG(BC)], [MGG], [AAG(BC)], [KFS],           !
1294    7!                                 [KSS], [UAF], [YS(BC)], [PNSF(BC)],           !
1295    7!                                 [PGMN(BC)], [PFJK], NGDR, USET(BC),           !
1296    7!                                 OGRIDLOD );                                   !
1297    7!              ELSE                                                             !
1298    7!                 IF SYM = -1 THEN                                              !
1299    8!$********************************* TAKEN OUT FOR ZAERO ***********************$! <-
1300    8!$                   CALL OFPALOAD ( NITER, BC, MINDEX, SUB, GSIZE, BGPDT(BC), $!
1301    8!$                                   [GTKG], [GSTKG], QDP, [AIRFRC(MINDEX)],   $!
1302    8!$                                   [DELTA(SUB)], [AAICMAT(MINDEX)],          $!
1303    8!$                                   [UAG(BC)], [MGG], [AAG(BC)], [KFS],       $!
1304    8!$                                   [KSS], [UAF], [YS(BC)], [PNSF(BC)],       $!
1305    8!$                                   [PGMN(BC)], [PFJK], NGDR, USET(BC),       $!
1306    8!$                                   OGRIDLOD );                               $!
1307    8!$*************************************************************************** $! <-
1308    8!                    CALL OFPALOAD ( NITER, BC, MINDEX, SUB, GSIZE, BGPDT(BC),  !
1309    8!                                    [UGTKG], [UGTKG], QDP, [AIRFRC(MINDEX)],   ! <-
1310    8!                                    [DELTA(SUB)], [AAICMAT(MINDEX)],           !
1311    8!                                    [UAG(BC)], [MGG], [AAG(BC)], [KFS],        !
1312    8!                                    [KSS], [UAF], [YS(BC)], [PNSF(BC)],        !
1313    8!                                    [PGMN(BC)], [PFJK], NGDR, USET(BC),        !
1314    8!                                    OGRIDLOD );                                !
1315    8!                 ENDIF;                                                        !
1316    7!              ENDIF;                                                           !
1317    6!$                                                                             $!
1318    6!$             CALL TO COMPUTE THE TRIMMED LOADS/DISPLACEMENTS ON THE          $!
1319    6!$             AERODYNAMIC MODEL                                               $!
1320    6!$                                                                             $!
1321    6!              IF SYM = 1  THEN                                                !
1322    7!$********************************* TAKEN OUT FOR ZAERO ***********************$! <-
1323    7!$                CALL OFPAEROM ( NITER, BC, MINDEX, SUB, GSIZE, GEOMSA,       $!
1324    7!$                                [GTKG], [GSTKG], QDP, [AIRFRC(MINDEX)],      $!
1325    7!$                                [DELTA(SUB)], [AICMAT(MINDEX)],               $!
1326    7!$                                [UAG(BC)], OAGRDLOD, OAGRDDSP );              $!
1327    7!$*************************************************************************** $! <-
1328    7!                 CALL OFPAEROM ( NITER, BC, MINDEX, SUB, GSIZE, GEOMSA,       !
1329    7!                                 [UGTKG], [UGTKG], QDP, [AIRFRC(MINDEX)],      !
1330    7!                                 [DELTA(SUB)], [AICMAT(MINDEX)],               !
1331    7!                                 [UAG(BC)], OAGRDLOD, OAGRDDSP );              !
1332    7!              ELSE                                                             !
1333    7!                 IF SYM = -1 THEN                                              !
1334    8!$********************************* TAKEN OUT FOR ZAERO ***********************$! <-
1335    8!$                   CALL OFPAEROM ( NITER, BC, MINDEX, SUB, GSIZE, GEOMSA,    $!
1336    8!$                                   [GTKG], [GSTKG], QDP, [AIRFRC(MINDEX)],   $!
1337    8!$                                   [DELTA(SUB)], [AAICMAT(MINDEX)],          $!
1338    8!$                                   [UAG(BC)], OAGRDLOD, OAGRDDSP );          $!
1339    8!$*************************************************************************** $! <-
1340    8!                    CALL OFPAEROM ( NITER, BC, MINDEX, SUB, GSIZE, GEOMSA,     !
1341    8!                                    [UGTKG], [UGTKG], QDP, [AIRFRC(MINDEX)],   ! <-
1342    8!                                    [DELTA(SUB)], [AAICMAT(MINDEX)],           !
1343    8!                                    [UAG(BC)], OAGRDLOD, OAGRDDSP );           !
1344    8!                 ENDIF;                                                        !
1345    7!              ENDIF;                                                           !
1346    6!           ENDDO;                                                              !
1347    5!        ENDIF;                                                                 !
1348    4!        IF BDRSP <> 0  THEN                                                    !
1349    5!           CALL OFPDLOAD ( NITER, BC, BGPDT(BC), PSIZE(BC), ESIZE(BC),       !
1350    5!                           [PHIG(BC)], [PTGLOAD], [PTHLOAD], [PFGLOAD],      !
1351    5!                           [PFHLOAD], OGRIDLOD );                            !
1352    5!           IF BDTR   <> 0 OR BMTR <> 0                                       !
1353    6!                CALL OFPSPCF ( NITER, BC, 5, 1, GSIZE, ESIZE(BC),           !
1354    6!                               NGDR, [KFS], , [UTRANF], , ,                 !
1355    6!                               [PNSF(BC)], [PGMN(BC)], [PFJK],              !
1356    6!                               [PHIG(BC)], [PTGLOAD], [PTHLOAD],            !
1357    6!                               BGPDT(BC), OGRIDLOD );                       !
1358    5!           IF BDFR   <> 0 OR BMFR <> 0                                       !
1359    6!                CALL OFPSPCF ( NITER, BC, 6, 2, GSIZE, ESIZE(BC),           !
1360    6!                               NGDR, [KFS], , [UFREQF], , ,                 !
```

```
1361    6!                                          [PNSF(BC)], [PGMN(BC)], [PFJK],      !
1362    6!                                          [PHIG(BC)], [PFGLOAD], [PFHLOAD],    !
1363    6!                                          BGPDT(BC), OGRIDLOD );               !
1364    5!          ENDIF;                                                               !
1365    4!          CALL OFPLOAD ( NUMOPTBC, BC, NITER, GSIZE, BGPDT(BC), PSIZE(BC),     !
1366    4!                         [PG] );                                               !
1367    4!          CALL OFPDISP ( NUMOPTBC, BC, NITER, GSIZE, BGPDT(BC), ESIZE(BC),     !
1368    4!                         PSIZE(BC), OGRIDDSP, [UG(BC)], [AG(BC)], [UAG(BC)],   !
1369    4!                         [AAG(BC)], [UBLASTG], , [UTRANG], [UTRANE], [UFREQG],  !
1370    4!                         [UFREQE], LAMBDA, [PHIG(BC)] );                       !
1371    4!          CALL EDR ( NUMOPTBC, BC, NITER, NDV, GSIZE, EOSUMMRY, EODISC,        !
1372    4!                     GLBDES, LOCLVAR, [PTRANS],                                !
1373    4!                     [UG(BC)], [UAG(BC)], , [UTRANG], [UFREQG], [PHIG(BC)] );  !
1374    4!          CALL PBKLEVAL ( BC, NITER, NDV, GLBDES, LOCLVAR, [PTRANS], PDLIST,    !
1375    4!                          OPNLBUCK );                                          !
1376    4!          CALL EBKLEVAL ( BC, NITER, NDV, GLBDES, LOCLVAR, [PTRANS], OEULBUCK ); !
1377    4!          CALL OFPEDR ( BC, HSIZE(BC), NITER );                                !
1378    4!       ENDDO;                                                                  !
1379    3!$                                                                            $!
1380    3!$                     SELECT ACTIVE CONSTRAINTS                              $!
1381    3!$                                                                            $!
1382    3!       PRINT("LOG=('          SENSITIVITY ANALYSIS')");                      !
1383    3!       CALL ACTCON ( NITER, MAXITER,  NRFAC, NDV, GLBDES, LOCLVAR, [PTRANS], !
1384    3!                     EPS,   APPCNVRG, GLBCNVRG,                              !
1385    3!                     CTL,   CTLMIN,   CONST, [AMAT], DESHIST, PFLAG, OLOCALDV ); !
1386    3!       CALL DESPUNCH ( NITER, PFLAG, OLOCALDV );                             !
1387    3!$                                                                            $!
1388    3!       IF GLBCNVRG OR NITER > MAXITER  THEN                                  !
1389    4!$                                                                           $!
1390    4!$         LAST ITERATION OUTPUT                                              $!
1391    4!$                                                                           $!
1392    4!          FOR BC = 1 TO NUMOPTBC DO                                         !
1393    5!             CALL OFPMROOT ( NITER, BC, NUMOPTBC, LAMBDA, 1 );              !
1394    5!             CALL OFPDISP ( NUMOPTBC, BC, NITER, GSIZE, BGPDT(BC), ESIZE(BC), !
1395    5!                            PSIZE(BC), OGRIDDSP,,,,,,,,,,LAMBDA,,, 1 );     !
1396    5!             CALL OFPEDR ( BC, HSIZE(BC), NITER, 1 );                       !
1397    5!          ENDDO;                                                            !
1398    4!       ENDIF;                                                               !
1399    3!$                                                                           $!
1400    3!       IF NOT GLBCNVRG AND NITER <= MAXITER THEN                            !
1401    4!$                                                                          $!
1402    4!$         USE APPROPRIATE RESIZING METHOD                                  $!
1403    4!$                                                                          $!
1404    4!          IF NITER >=  FSDS AND NITER <= FSDE THEN                         !
1405    5!             CALL FSD ( NDV, NITER, FSDS, FSDE, MPS, OCS, ALPHA,           !
1406    5!                        CNVRGLIM, GLBDES, LOCLVAR, [PTRANS], CONST,        !
1407    5!                        APPCNVRG, CTL, CTLMIN, DESHIST );                  !
1408    5!          ENDIF;                                                           !
1409    4!$                                                                          $!
1410    4!          IF ( NITER >= MPS AND NITER <= MPE ) OR                          !
1411    5!             ( NITER >= OCS AND NITER <= OCE ) THEN                        !
1412    5!$                                                                         $!
1413    5!$            USE MATHEMATICAL PROGRAMMING OR OC METHODS                   $!
1414    5!$                                                                         $!
1415    5!$            OBTAIN THE SENSITIVITIES OF THE CONSTRAINTS WRT THE          $!
1416    5!$            DESIGN VARIABLES                                             $!
1417    5!$                                                                         $!
1418    5!             CALL MAKDFV ( NITER, NDV, [PMINT], [PMAXT], CONST, [AMAT] );  !
1419    5!             CALL LAMINSNS ( NITER, NDV, GLBDES, LOCLVAR, [PTRANS], CONST,  !
1420    5!                             [AMAT] );                                      !
1421    5!$                                                                         $!
1422    5!$*************************************************************************$!
1423    5!$            SENSITIVITY EVALUATION FOR BOUNDARY CONDITION DEPENDENT CONSTRAINTS$!
1424    5!$*************************************************************************$!
1425    5!$                                                                         $!
1426    5!             FOR BC = 1 TO NUMOPTBC DO                                     !
1427    6!                CALL ABOUND ( NITER, BC, CONST, ACTBOUND, NAUS, NACSD, [PGAS], !
1428    6!                              PCAS, ACTAERO, ACTDYN, ACTFLUT, ACTPNL, ACTBAR, !
1429    6!                              NMPC, NSPC, NOMIT, NRSET, NGDR, USET(BC) );     !
1430    6!                IF ACTBOUND THEN                                            !
1431    7!$                                                                          $!
1432    7!$                  REESTABLISH THE BASE USET AND PARTITIONING DATA FOR THE BC $!
1433    7!$                  IF GDR CHANGED IT                                       $!
1434    7!$                  NOTE, THIS LEAVES AN INCOMPATIBILITY BETWEEN USET(BC) AND $!
1435    7!$                  BGPDT(BC) SINCE THE LATTER IS NOT REGENERATED.          $!
1436    7!$                  THIS INCOMPATIBILITY WILL NOT AFFECT THE SENSITIVITY ANALYSIS$!
1437    7!$                  AND WILL BE CORRECTED IN THE SUBSEQUENT ANALYSIS        $!
1438    7!$                                                                          $!
1439    7!                   IF NGDR <> 0 THEN                                       !
1440    8!                      CALL MKUSET(BC, GSIZEB, [YS(BC)], [TMN(BC)], [PGMN(BC)], !
1441    8!                                  [PNSF(BC)], [PFOA(BC)], [PARL(BC)], USET(BC)); !
```

```
1442    8!                ENDIF;                                                                  !
1443    7!$                                                                                       $!
1444    7!$               EVALUATE FREQUENCY CONSTRAINT SENSITIVITIES                             $!
1445    7!$                                                                                       $!
1446    7!                IF ACTDYN THEN                                                          !
1447    8!                   IF NGDR <> 0 THEN                                                    !
1448    9!                      CALL ROWPART ( [PHIG(BC)], , [GTMP], [PGDRG(BC)] );               !
1449    9!                      CALL FREQSENS ( NITER, BC, NDV, GLBDES, CONST, LAMBDA,            !
1450    9!                                      GMKCT, DKVI, GMMCT, DMVI,                         !
1451    9!                                      [GTMP], [AMAT] );                                 !
1452    9!                   ELSE                                                                 !
1453    9!                      CALL FREQSENS ( NITER, BC, NDV, GLBDES, CONST, LAMBDA,            !
1454    9!                                      GMKCT, DKVI, GMMCT, DMVI,                         !
1455    9!                                      [PHIG(BC)], [AMAT] );                             !
1456    9!                   ENDIF;                                                               !
1457    8!                ENDIF;                                                                  !
1458    7!$                                                                                       $!
1459    7!$               EVALUATE FLUTTER CONSTRAINT SENSITIVITIES                               $!
1460    7!$                                                                                       $!
1461    7!                IF ACTFLUT THEN                                                         !
1462    8!                   SUB  := 0;                                                           !
1463    8!                   LOOP := TRUE;                                                        !
1464    8!                   IF NGDR <> 0 CALL ROWPART ([PHIG(BC)],,[GTMP],[PGDRG(BC)]);          !
1465    8!                   WHILE LOOP DO                                                        !
1466    9!                      SUB := SUB + 1;                                                   !
1467    9!                      IF NGDR <> 0 THEN                                                 !
1468   10!                         CALL FLUTSENZ (NITER, BC, SUB, LOOP, GSIZEB, NDV,              ! ←
1469   10!                                        GLBDES, CONST, GMKCT, DKVI, GMMCT,              !
1470   10!                                        DMVI, CLAMBDA, LAMBDA,                          !
1471   10!                                        [QHHLFL(BC,SUB)],                               !
1472   10!                                        [MHHFL(BC,SUB)], [BHHFL(BC,SUB)],               !
1473   10!                                        [KHHFL(BC,SUB)], [GTMP], [AMAT],                !
1474   10!                                        AEROZ );                                        ! ←
1475   10!                      ELSE                                                              !
1476   10!                         CALL FLUTSENZ (NITER, BC, SUB, LOOP, GSIZEB, NDV,              ! ←
1477   10!                                        GLBDES, CONST, GMKCT, DKVI, GMMCT,              !
1478   10!                                        DMVI, CLAMBDA, LAMBDA,                          !
1479   10!                                        [QHHLFL(BC,SUB)],                               !
1480   10!                                        [MHHFL(BC,SUB)], [BHHFL(BC,SUB)],               !
1481   10!                                        [KHHFL(BC,SUB)], [PHIG(BC)], [AMAT],            !
1482   10!                                        AEROZ );                                        ! ←
1483   10!                      ENDIF;                                                            !
1484    9!                   ENDDO;                                                               !
1485    8!                ENDIF;                                                                  !
1486    7!$                                                                                       $!
1487    7!$               EVALUATE ACTIVE DISPLACEMENT DEPENDENT CONSTRAINTS FROM                 $!
1488    7!$               THE STATICS DISCIPLINE                                                  $!
1489    7!$                                                                                       $!
1490    7!                IF NAUS > 0 THEN                                                        !
1491    8!$                                                                                       $!
1492    8!$                  SENSITIVITIES OF CONSTRAINTS WRT DISPLACEMENTS FOR STATICS            $!
1493    8!$                                                                                       $!
1494    8!                   CALL NULLMAT ( [DFDU], [DPGV] );                                     !
1495    8!                   IF NACSD > NAUS * NDV THEN                                           !
1496    9!$                                                                                       $!
1497    9!$                     USE GRADIENT METHOD                                               $!
1498    9!$                                                                                       $!
1499    9!                      CALL MAKDFU ( NITER, BC, GSIZEB, [SMAT], [GLBSIG],               !
1500    9!                                    CONST, [DFDU] );                                   !
1501    9!                   ELSE                                                                 !
1502    9!$                                                                                       $!
1503    9!$                     USE VIRTUAL LOAD METHOD                                           $!
1504    9!$                                                                                       $!
1505    9!                      CALL MAKDFU ( NITER, BC, GSIZEB, [SMAT], [GLBSIG],               !
1506    9!                                    CONST, [DPGV] );                                   !
1507    9!                   ENDIF;                                                               !
1508    8!$                                                                                       $!
1509    8!$                  SOME RELATIVELY SIMPLE CALCULATIONS THAT PRECEDE THE                 $!
1510    8!$                  LOOP ON THE DESIGN VARIABLES                                         $!
1511    8!$                                                                                       $!
1512    8!                   IF NGDR <> 0 THEN                                                    !
1513    9!                      CALL PARTN ( [UG(BC)],,,, [UGA], [PGAS], [PGDRG(BC)]);            !
1514    9!                   ELSE                                                                 !
1515    9!                      CALL COLPART ( [UG(BC)], , [UGA], [PGAS] );                       !
1516    9!                   ENDIF;                                                               !
1517    8!$                                                                                       $!
1518    8!$                  OBTAIN THE SENSITIVITIES OF THE DESIGN                               $!
1519    8!$                  DEPENDENT LOADS                                                      $!
1520    8!$                                                                                       $!
1521    8!                   CALL DDLOAD(NDV, GSIZEB, BC, SMPLOD, DDFLG,[PGAS],[DPVJ]);          !
1522    8!$                                                                                       $!
```

```
1523    8!                CALL MAKDVU ( NITER, NDV, GLBDES, [UGA], [DKUG],                !
1524    8!                            GMKCT, DKVI );                                       !
1525    8!                CALL NULLMAT ( [DUG] );                                          !
1526    8!                IF NRSET <> 0 THEN                                               !
1527    9!                   IF NGDR <> 0 THEN                                             !
1528   10!                      CALL PARTN ([AG(BC)],,,, [AGA], [PGAS], [PGDRG(BC)]);      !
1529   10!                   ELSE                                                          !
1530   10!                      CALL COLPART ( [AG(BC)], , [AGA], [PGAS] );                !
1531   10!                   ENDIF;                                                        !
1532    9!                   CALL MAKDVU ( NITER, NDV, GLBDES, [AGA], [DMAG],             !
1533    9!                               GMMCT, DMVI );                                    !
1534    9!                   [DUG] := [DKUG] + [DMAG];                                     !
1535    9!                ELSE                                                             !
1536    9!                   [DUG] := [DKUG];                                              !
1537    9!                ENDIF;                                                           !
1538    8!$                                                                              $!
1539    8!$               ACCOUNT FOR VIRTUAL LOAD METHOD                                $!
1540    8!$                                                                              $!
1541    8!                IF NACSD > NAUS * NDV THEN                                      !
1542    9!$                                                                             $!
1543    9!$                  USE GRADIENT METHOD                                         $!
1544    9!$                                                                             $!
1545    9!                   IF DDFLG > 0 THEN                                           !
1546   10!                      [DPGV] := [DPVJ] + [DUG];                                !
1547   10!                   ELSE                                                        !
1548   10!                      [DPGV] := [DUG];                                         !
1549   10!                   ENDIF;                                                      !
1550    9!                ELSE                                                           !
1551    9!$                                                                            $!
1552    9!$                  USE VIRTUAL LOAD METHOD                                     $!
1553    9!$                                                                             $!
1554    9!                   IF DDFLG > 0 THEN                                           !
1555   10!                      [DFDU] := [DPVJ] + [DUG];                                !
1556   10!                   ELSE                                                        !
1557   10!                      [DFDU] := [DUG];                                         !
1558   10!                   ENDIF;                                                      !
1559    9!                ENDIF;                                                         !
1560    8!$                                                                            $!
1561    8!$               REDUCE THE RIGHT HAND SIDES TO THE L SET                       $!
1562    8!$                                                                             $!
1563    8!                CALL NULLMAT ( [DPNV], [DMUN] );                               !
1564    8!                IF NMPC <> 0 THEN                                              !
1565    9!                   CALL GREDUCE (,[DPGV], [PGMNS(BC)],[TMN(BC)],, [DPNV]);     !
1566    9!                ELSE                                                           !
1567    9!                   [DPNV] := [DPGV];                                           !
1568    9!                ENDIF;                                                         !
1569    8!$                                                                            $!
1570    8!                CALL NULLMAT ( [DPFV], [DMUF] );                               !
1571    8!                IF NSPC <> 0 THEN                                              !
1572    9!                   CALL NREDUCE (, [DPNV], [PNSFS(BC)], , , , , [DPFV]);       !
1573    9!                ELSE                                                           !
1574    9!                   [DPFV] := [DPGV];                                           !
1575    9!                ENDIF;                                                         !
1576    8!$                                                                            $!
1577    8!                CALL NULLMAT ( [DPAV], [DMUA] );                               !
1578    8!                IF NGDR <> 0 THEN                                              !
1579    9!                   [DPAV] := TRANS( [GSUBO(BC)] ) * [DPFV];                    !
1580    9!                ELSE                                                           !
1581    9!                   IF NOMIT <> 0 THEN                                          !
1582   10!                      CALL FREDUCE (, [DPFV], [PFOAS(BC)], ,                   !
1583   10!                                    [KOOINV(BC)], , , [GSUBO(BC)], ,           !
1584   10!                                    [DPAV], [DPOV], );                         !
1585   10!                   ELSE                                                        !
1586   10!                      [DPAV] := [DPFV];                                        !
1587   10!                   ENDIF;                                                      !
1588    9!                ENDIF;                                                         !
1589    8!$                                                                            $!
1590    8!                IF NRSET <> 0 THEN                                             !
1591    9!                   CALL ROWPART ( [DPAV], [DPRV], [DPLV], [PARLS(BC)] );       !
1592    9!                   [DRHS] := TRANS( [D(BC)] ) * [DPLV] + [DPRV];               !
1593    9!$                                                                            $!
1594    9!$                  PROCESS ACTIVE CONSTRAINTS FOR STATICS DISCIPLINE          $!
1595    9!$                                                                            $!
1596    9!                   CALL INERTIA ( [MRR(BC)], [DRHS], [DURD] );                 !
1597    9!                   [DULD] := [D(BC)] * [DURD];                                 !
1598    9!                   CALL ROWMERGE ( [DUAD], [DURD], [DULD], [PARLS(BC)] );      !
1599    9!                   [DPLV] := [DPLV] + [IFR(BC)] * [DURD];                      !
1600    9!                   CALL FBS ( [KLLINV(BC)], [DPLV], [DULV] );                  !
1601    9!                   CALL YSMERGE ( [DUAV], , [DULV], [PARLS(BC)] );             !
1602    9!                ELSE                                                           !
1603    9!                   CALL FBS ( [KLLINV(BC)], [DPAV], [DUAV] );                  !
```

```
1604    9!                    ENDIF;                                                    !
1605    8!$                                                                           $!
1606    8!$                   RECOVER TO THE F SET                                    $!
1607    8!$                                                                           $!
1608    8!                    CALL NULLMAT ( [DUFV] );                                 !
1609    8!                    IF NGDR <> 0 THEN                                        !
1610    9!                       [DUFV] := [GSUBO(BC)] * [DUAV];                       !
1611    9!                    ELSE                                                     !
1612    9!                       IF NOMIT <> 0 THEN                                    !
1613   10!                          IF NRSET <> 0 THEN                                 !
1614   11!                             [TMP1] := [DPOV] - [IFM(BC)] * [DUAD];          !
1615   11!                          ELSE                                               !
1616   11!                             [TMP1] := [DPOV];                               !
1617   11!                          ENDIF;                                             !
1618   10!                          CALL FBS ( [KOOINV(BC)], [TMP1], [UOO] );          !
1619   10!                          [UO] := [GSUBO(BC)] * [DUAV] + [UOO];              !
1620   10!                          CALL ROWMERGE ([DUFV], [UO], [DUAV], [PFOAS(BC)]); !
1621   10!                       ELSE                                                  !
1622   10!                          [DUFV] := [DUAV];                                  !
1623   10!                       ENDIF;                                                !
1624    9!                    ENDIF;                                                   !
1625    8!$                                                                          $!
1626    8!$                   REDUCE THE LEFT HAND SIDE MATRIX                       $!
1627    8!$                                                                          $!
1628    8!                    IF NMPC <> 0 THEN                                        !
1629    9!                       CALL GREDUCE (,[DFDU],[PGMNS(BC)],[TMN(BC)],,[DFDUN]); !
1630    9!                    ELSE                                                     !
1631    9!                       [DFDUN] := [DFDU];                                    !
1632    9!                    ENDIF;                                                   !
1633    8!$                                                                          $!
1634    8!                    IF NSPC <> 0 THEN                                        !
1635    9!                       CALL ROWPART ( [DFDUN], , [DFDUF], [PNSFS(BC)] );     !
1636    9!                    ELSE                                                     !
1637    9!                       [DFDUF] := [DFDUN];                                   !
1638    9!                    ENDIF;                                                   !
1639    8!$                                                                          $!
1640    8!$                   ACCOUNT FOR VIRTUAL LOAD METHOD                        $!
1641    8!$                                                                          $!
1642    8!                    IF NACSD > NAUS * NDV THEN                               !
1643    9!$                                                                          $!
1644    9!$                      USE GRADIENT METHOD                                 $!
1645    9!$                                                                          $!
1646    9!                       CALL MKAMAT ([AMAT], [DFDUF], [DUFV], PCAS, [PGAS] ); !
1647    9!                    ELSE                                                     !
1648    9!$                                                                          $!
1649    9!$                      USE VIRTUAL LOAD METHOD                             $!
1650    9!$                                                                          $!
1651    9!                       CALL MKAMAT ([AMAT], [DUFV], [DFDUF], PCAS, [PGAS] ); !
1652    9!                    ENDIF;                                                   !
1653    8!$                                                                          $!
1654    8!                 ENDIF;    $   END IF ON ACTIVE APPLIED STATIC LOADS       $!
1655    7!$                                                                          $!
1656    7!$                EVALUATE ACTIVE CONSTRAINTS FROM                          $!
1657    7!$                THE STATIC AEROELASTICITY DISCIPLINE                      $!
1658    7!$                                                                          $!
1659    7!                 IF ACTAERO THEN                                            !
1660    8!                    LOOP    := TRUE;                                        !
1661    8!                    ACTUAGG := FALSE;                                       !
1662    8!                    SUB     := 0;                                           !
1663    8!                    CALL NULLMAT ( [DUFV] );                                !
1664    8!                    WHILE LOOP DO                                           !
1665    9!                       SUB := SUB + 1;                                      !
1666    9!                       CALL AROSNSDR ( NITER, BC, SUB, LOOP, MINDEX, CONST, !
1667    9!                                       SYM, NGDR,                          !
1668    9!                                       [PGDRG(BC)], [UAG(BC)], [AAG(BC)],  !
1669    9!                                       ACTUAG, [UGA], [AGA], [PGAA], [PGAU], !
1670    9!                                       PCAA,[UAGC(BC,SUB)], [AAGC(BC,SUB)], !
1671    9!                                       ACTAEFF,[AUAGC], [AAAGC], PCAE );   !
1672    9!                       IF ACTAEFF THEN                                      !
1673   10!$                                                                          $!
1674   10!$                      PROCESS PSEUDO DISPLACEMENTS FOR EFFECTIVENESS      $!
1675   10!$                      CONSTRAINTS                                         $!
1676   10!$                                                                          $!
1677   10!                       CALL MAKDVU ( NITER, NDV, GLBDES, [AUAGC], [DKUG],  !
1678   10!                                     GMKCT, DKVI );                        !
1679   10!                       IF NRSET <> 0 THEN                                   !
1680   11!                          CALL MAKDVU ( NITER, NDV, GLBDES, [AAAGC], [DMAG],!
1681   11!                                        GMMCT, DMVI);                       !
1682   11!                          [DPGV] := [DKUG] + [DMAG];                        !
1683   11!                          CALL MAKDVU ( NITER, NDV, GLBDES, [AUAGC], [DMUG],!
1684   11!                                        GMMCT, DMVI);                       !
```

```
1685   11!                ELSE                                                    !
1686   11!                   [DPGV] := [DKUG];                                    !
1687   11!                ENDIF;                                                  !
1688   10!$                                                                      $!
1689   10!$               REDUCE THE RIGHT HAND SIDES TO THE L SET                $!
1690   10!$                                                                      $!
1691   10!                CALL NULLMAT ( [DPNV], [DMUN] );                        !
1692   10!                IF NMPC <> 0 THEN                                       !
1693   11!                   CALL GREDUCE ( , [DPGV], [PGMNS(BC)], [TMN(BC)],,    !
1694   11!                                  [DPNV]);                              !
1695   11!                   IF NRSET <> 0 CALL GREDUCE ( , [DMUG],               !
1696   12!                                 [PGMNS(BC)], [TMN(BC)],, [DMUN] );     !
1697   11!                ELSE                                                    !
1698   11!                   [DPNV] := [DPGV];                                    !
1699   11!                   IF NRSET <> 0 [DMUN] := [DMUG];                      !
1700   11!                ENDIF;                                                  !
1701   10!$                                                                      $!
1702   10!                CALL NULLMAT ( [DPFV], [DMUF] );                        !
1703   10!                IF NSPC <> 0 THEN                                       !
1704   11!                   CALL NREDUCE (,[DPNV],[PNSFS(BC)],,,,, [DPFV]);      !
1705   11!                   IF NRSET <> 0                                        !
1706   12!                      CALL NREDUCE (,[DMUN],[PNSFS(BC)],,,,, [DMUF]);   !
1707   11!                ELSE                                                    !
1708   11!                   [DPFV] := [DPGV];                                    !
1709   11!                   IF NRSET <> 0 [DMUF] := [DMUN];                      !
1710   11!                ENDIF;                                                  !
1711   10!$                                                                      $!
1712   10!                CALL NULLMAT ( [DPAV], [DMUA] );                        !
1713   10!                IF NGDR <> 0 THEN                                       !
1714   11!                   [DPAV] := TRANS( [GSUBO(BC)] ) * [DPFV];             !
1715   11!                   IF NRSET <> 0 [DMUA]:=TRANS([GSUBO(BC)])*[DMUF];     !
1716   11!                ELSE                                                    !
1717   11!                   IF NOMIT <> 0 THEN                                   !
1718   12!                      CALL FREDUCE ( , [DPFV], [PFOAS(BC)], 1,          !
1719   12!                                     [KOOL(BC,SUB)], [KOOU(BC,SUB)],    !
1720   12!                                     [KAO(BC,SUB)], [GASUBO(BC,SUB)], ,!
1721   12!                                     [DPAV], [DPOV], );                 !
1722   12!                      IF NRSET <> 0                                     !
1723   13!                         CALL FREDUCE ( , [DMUF], [PFOAS(BC)], 1,       !
1724   13!                                        [KOOL(BC,SUB)], [KOOU(BC,SUB)], !
1725   13!                                        [KAO(BC,SUB)], [GASUBO(BC,SUB)], ,!
1726   13!                                        [DMUA], [DMUO], );              !
1727   12!                   ELSE                                                 !
1728   12!                      [DPAV] := [DPFV];                                 !
1729   12!                      IF NRSET <> 0  [DMUA] := [DMUF];                  !
1730   12!                   ENDIF;                                               !
1731   11!                ENDIF;                                                  !
1732   10!$                                                                      $!
1733   10!                IF NRSET <> 0 THEN                                      !
1734   11!                   CALL ROWPART ([DPAV],[DPRV],[DPLV],[PARLS(BC)] ); !
1735   11!                   CALL ROWPART ([DMUA],[DMUR],[DMUL],[PARLS(BC)] ); !
1736   11!                   CALL GFBS ( [RL11(BC,SUB)], [RU11(BC,SUB)],         !
1737   11!                               [DPLV], [R11DPL] );                      !
1738   11!                   [DP1]  := TRANS([D(BC)]) * [DMUL] + [DMUR] -         !
1739   11!                             [R21(BC,SUB)] * [R11DPL];                  !
1740   11!                   [DRHS] := TRANS( [D(BC)] ) * [DPLV] + [DPRV] -       !
1741   11!                             [R31(BC,SUB)] * [R11DPL];                  !
1742   11!$                                                                     $!
1743   11!$                   PROCESS ACTIVE CONSTRAINTS FOR SAERO DISCIPLINE   $!
1744   11!$                                                                     $!
1745   11!                   CALL GFBS ( [KL11(BC,SUB)], [KU11(BC,SUB)],        !
1746   11!                               [DP1],          [DK1V] );                !
1747   11!                   [DRHS] := [DRHS] - [K21(BC,SUB)] * [DK1V];           !
1748   11!$                                                                     $!
1749   11!                   CALL DECOMP ( [LHSA(BC,SUB)], [LHSL], [LHSU] );    !
1750   11!                   CALL GFBS ( [LHSL], [LHSU], [DRHS], [DU2] );       !
1751   11!$                                                                     $!
1752   11!                   [DU1R]     := [DK1V] + [K1112(BC,SUB)] * [DU2];     !
1753   11!                   [DU1L]     := [R11DPL] + [R1112(BC,SUB)] * [DU1R] +!
1754   11!                                            [R1113(BC,SUB)] * [DU2];   !
1755   11!                   [EFFSENS] := - [R31(BC,SUB)] * [DU1L] -              !
1756   11!                                [R32(BC,SUB)] * [DU1R];                 !
1757   11!$                                                                     $!
1758   11!                   CALL AEROEFFS ( NITER, BC, SUB, SYM, NDV, CONST,   !
1759   11!                                   PCAE, [EFFSENS], [AMAT] );         !
1760   11!                ELSE                                                    !
1761   11!$                                                                     $!
1762   11!$                   NOTE THAT SAERO W/O SUPPORT IS NOT SUPPORTED      $!
1763   11!$                                                                     $!
1764   11!                ENDIF;                                                  !
1765   10!             ENDIF; $ END IF ON ACTAEFF                                $!
```

```
1766    9!$                                                                $!
1767    9!              IF ACTUAG THEN                                      !
1768   10!$                                                                 $!
1769   10!$                 SENSITIVITIES OF CONSTRAINTS WRT DISPLACEMENTS  $!
1770   10!$                 FOR SAERO.  THE ACTUAGG FLAG WILL BE RETURNED   $!
1771   10!$                 FALSE IF ONLY TRIM PARAMETER CONSTRAINTS ARE ACTIVE $!
1772   10!$                                                                 $!
1773   10!                  CALL NULLMAT ( [DFDU] );                        !
1774   10!                  CALL MAKDFU ( NITER, BC, GSIZEB, [SMAT], [GLBSIG],  !
1775   10!                                CONST, [DFDU], ACTUAGG, SUB );    !
1776   10!$                                                                 $!
1777   10!$                 SOME RELATIVELY SIMPLE CALCULATIONS THAT PRECEDE $!
1778   10!$                 THE LOOP ON THE DESIGN VARIABLES                $!
1779   10!$                                                                 $!
1780   10!                  CALL MAKDVU ( NITER, NDV, GLBDES, [UGA], [DKUG],  !
1781   10!                                GMKCT, DKVI );                    !
1782   10!                  CALL NULLMAT ( [DPGV] );                        !
1783   10!                  IF NRSET <> 0 THEN                              !
1784   11!                     CALL MAKDVU ( NITER, NDV, GLBDES, [AGA], [DMAG],  !
1785   11!                                   GMMCT, DMVI );                 !
1786   11!                     [DPGV] := [DKUG] + [DMAG];                   !
1787   11!                     CALL MAKDVU ( NITER, NDV, GLBDES, [UGA], [DMUG],  !
1788   11!                                   GMMCT, DMVI );                 !
1789   11!                  ELSE                                            !
1790   11!                     [DPGV] := [DKUG];                            !
1791   11!                  ENDIF;                                          !
1792   10!$                                                                 $!
1793   10!$                 REDUCE THE RIGHT HAND SIDES TO THE L SET        $!
1794   10!$                                                                 $!
1795   10!                  CALL NULLMAT ( [DPNV], [DMUN] );                !
1796   10!                  IF NMPC <> 0 THEN                               !
1797   11!                     CALL GREDUCE ( , [DPGV], [PGMNS(BC)], [TMN(BC)],, !
1798   11!                                    [DPNV]);                      !
1799   11!                     IF NRSET <> 0 CALL GREDUCE ( , [DMUG],       !
1800   12!                                   [PGMNS(BC)], [TMN(BC)],, [DMUN] );  !
1801   11!                  ELSE                                            !
1802   11!                     [DPNV] := [DPGV];                            !
1803   11!                     IF NRSET <> 0 [DMUN] := [DMUG];              !
1804   11!                  ENDIF;                                          !
1805   10!$                                                                 $!
1806   10!                  CALL NULLMAT ( [DPFV], [DMUF] );                !
1807   10!                  IF NSPC <> 0 THEN                               !
1808   11!                     CALL NREDUCE (,[DPNV],[PNSFS(BC)],,,,, [DPFV]);  !
1809   11!                     IF NRSET <> 0                                !
1810   12!                        CALL NREDUCE (,[DMUN],[PNSFS(BC)],,,,, [DMUF]);!
1811   11!                  ELSE                                            !
1812   11!                     [DPFV] := [DPGV];                            !
1813   11!                     IF NRSET <> 0 [DMUF] := [DMUN];              !
1814   11!                  ENDIF;                                          !
1815   10!$                                                                 $!
1816   10!                  CALL NULLMAT ( [DPAV], [DMUA] );                !
1817   10!                  IF NGDR <> 0 THEN                               !
1818   11!                     [DPAV] := TRANS( [GSUBO(BC)] ) * [DPFV];     !
1819   11!                     IF NRSET <> 0 [DMUA]:=TRANS([GSUBO(BC)])*[DMUF];  !
1820   11!                  ELSE                                            !
1821   11!                     IF NOMIT <> 0 THEN                           !
1822   12!                        CALL FREDUCE ( , [DPFV], [PFOAS(BC)], 1,  !
1823   12!                                   [KOOL(BC,SUB)], [KOOU(BC,SUB)],  !
1824   12!                                   [KAO(BC,SUB)], [GASUBO(BC,SUB)], ,  !
1825   12!                                   [DPAV], [DPOV], );             !
1826   12!                        IF NRSET <> 0                             !
1827   13!                           CALL FREDUCE ( , [DMUF], [PFOAS(BC)], 1,  !
1828   13!                                   [KOOL(BC,SUB)], [KOOU(BC,SUB)],  !
1829   13!                                   [KAO(BC,SUB)], [GASUBO(BC,SUB)], ,  !
1830   13!                                   [DMUA], [DMUO], );             !
1831   12!                     ELSE                                         !
1832   12!                        [DPAV] := [DPFV];                         !
1833   12!                        IF NRSET <> 0  [DMUA] := [DMUF];          !
1834   12!                     ENDIF;                                       !
1835   11!                  ENDIF;                                          !
1836   10!$                                                                 $!
1837   10!                  IF NRSET <> 0 THEN                              !
1838   11!                     CALL ROWPART ([DPAV],[DPRV],[DPLV],[PARLS(BC)] ); !
1839   11!                     CALL ROWPART ([DMUA],[DMUR],[DMUL],[PARLS(BC)] ); !
1840   11!                     CALL GFBS ( [RL11(BC,SUB)], [RU11(BC,SUB)],  !
1841   11!                                 [DPLV], [R11DPL] );              !
1842   11!                     [DP1]  := TRANS([D(BC)]) * [DMUL] + [DMUR] -  !
1843   11!                               [R21(BC,SUB)] * [R11DPL];          !
1844   11!                     [DRHS] := TRANS( [D(BC)] ) * [DPLV] + [DPRV] -  !
1845   11!                               [R31(BC,SUB)] * [R11DPL];          !
1846   11!$                                                                 $!
```

```
1847   11!$                  PROCESS ACTIVE CONSTRAINTS FOR SAERO DISCIPLINE  $!
1848   11!$                                                                   $!
1849   11!                   CALL GFBS ( [KL11(BC,SUB)], [KU11(BC,SUB)],       !
1850   11!                               [DP1],          [DK1V] );           !
1851   11!                   [DRHS] := [DRHS] - [K21(BC,SUB)] * [DK1V];       !
1852   11!$                                                                   $!
1853   11!                   CALL AEROSENS ( NITER, BC, MINDEX, SUB, CONST,    !
1854   11!                                   SYM, NDV,                         !
1855   11!                                   BGPDT(BC), STABCF, [PGAA],        !
1856   11!                                   [LHSA(BC,SUB)], [RHSA(BC,SUB)],   !
1857   11!                                   [DRHS], [AAR], [DDELDV], [AMAT] );!
1858   11!$                                                                   $!
1859   11!                   [DURV] := [K1112(BC,SUB)] * [AAR] +               !
1860   11!                             [PAR(BC,SUB)] * [DDELDV] + [DK1V];      !
1861   11!                   [DULV] := [R1112(BC,SUB)] * [DURV] +              !
1862   11!                             [R1113(BC,SUB)] * [AAR]  -              !
1863   11!                             [R11PAL(BC,SUB)] * [DDELDV] + [R11DPL]; !
1864   11!                   CALL ROWMERGE ([DUAV],[DURV],[DULV],[PARLS(BC)]); !
1865   11!                ELSE                                                 !
1866   11!$                                                                   $!
1867   11!$                  NOTE THAT SAERO W/O SUPPORT IS NOT SUPPORTED      $!
1868   11!$                                                                   $!
1869   11!                ENDIF;                                               !
1870   10!$                                                                   $!
1871   10!$                RECOVER SENSITIVITIES TO THE F SET                 $!
1872   10!$                                                                   $!
1873   10!                 CALL NULLMAT ( [UAFTMP] );                         !
1874   10!                 IF NGDR <> 0 THEN                                  !
1875   11!                    [UAFTMP] := [GASUBO(BC,SUB)] * [DUAV];          !
1876   11!                 ELSE                                               !
1877   11!                    IF NOMIT <> 0 THEN                              !
1878   12!                       IF NRSET <> 0 THEN                           !
1879   13!                          [TMP1] := [DPOV]+[POARO(BC,SUB)]*[DDELDV]; !
1880   13!                       ELSE                                         !
1881   13!                          [TMP1] := [DPOV];                         !
1882   13!                       ENDIF;                                       !
1883   12!                       CALL GFBS ( [KOOL(BC,SUB)], [KOOU(BC,SUB)],  !
1884   12!                                   [TMP1], [UOO]);                  !
1885   12!                       [UO] := [GASUBO(BC,SUB)] * [DUAV] + [UOO];   !
1886   12!                       CALL ROWMERGE ( [UAFTMP], [UO], [DUAV],      !
1887   12!                                       [PFOAS(BC)] );               !
1888   12!                    ELSE                                            !
1889   12!                       [UAFTMP] := [DUAV];                          !
1890   12!                    ENDIF;                                          !
1891   11!                 ENDIF;                                             !
1892   10!                 CALL AROSNSMR ( BC, SUB, NDV, [PGAA], [PGAU], [DUFV],!
1893   10!                                 [UAFTMP] );                        !
1894   10!$                                                                  $!
1895   10!              ENDIF; $   END IF ON ACTUAG                          $!
1896    9!             ENDDO; $   END DO ON SUBSCRIPT LOOP                   $!
1897    8!$                                                                  $!
1898    8!             IF ACTUAGG THEN                                        !
1899    9!$                                                                  $!
1900    9!$               REDUCE THE LEFT HAND SIDE MATRIX                   $!
1901    9!$                                                                  $!
1902    9!               CALL NULLMAT ( [DFDUN] );                          !
1903    9!               IF NMPC <> 0 THEN                                  !
1904   10!                  CALL GREDUCE ( , [DFDU], [PGMNS(BC)], [TMN(BC)],, !
1905   10!                                 [DFDUN]);                        !
1906   10!               ELSE                                               !
1907   10!                  [DFDUN] := [DFDU];                              !
1908   10!               ENDIF;                                             !
1909    9!$                                                                 $!
1910    9!               CALL NULLMAT ( [DFDUF] );                          !
1911    9!               IF NSPC <> 0 THEN                                  !
1912   10!                  CALL ROWPART ( [DFDUN], , [DFDUF], [PNSFS(BC)] ); !
1913   10!               ELSE                                               !
1914   10!                  [DFDUF] := [DFDUN];                             !
1915   10!               ENDIF;                                             !
1916    9!$                                                                 $!
1917    9!$               TAKE MERGED SENSITIVITIES OF DISPLACEMENTS AND     $!
1918    9!$               COMPUTE THE AMAT MATRIX TERMS FOR THE SAERO        $!
1919    9!$               CONSTRAINTS                                        $!
1920    9!$                                                                 $!
1921    9!               CALL MKAMAT ([AMAT], [DFDUF], [DUFV], PCAA, [PGAU] );!
1922    9!$                                                                 $!
1923    9!            ENDIF; $   END IF ON ANY ACTIVE DISPLACEMENTS         $!
1924    8!          ENDIF;       $   END IF ON ACTIVE AEROELASTIC CONSTRAINTS $!
1925    7!$                                                                 $!
1926    7!$         EVALUATE PANEL BUCKLING CONSTRAINT SENSITIVITIES        $!
1927    7!$                                                                 $!
```

```
1928    7!                  IF ACTPNL THEN                                                  !
1929    8!                     CALL PBKLSENS ( BC, NITER, NDV, GLBDES, LOCLVAR, [PTRANS], !
1930    8!                                     PDLIST );                                    !
1931    8!                  ENDIF;                                                          !
1932    7!                  IF ACTBAR THEN                                                  !
1933    8!                     CALL EBKLSENS ( BC, NITER, NDV, GLBDES, LOCLVAR, [PTRANS]);!
1934    8!                  ENDIF;                                                          !
1935    7!               ENDIF;          $   END IF ON ACTIVE BOUNDARY CONDITION          $!
1936    6!            ENDDO;             $   END DO ON ACTIVE BOUNDARY CONDITIONS         $!
1937    5!$                                                                               $!
1938    5!            CALL OFPGRAD ( NITER, NUMOPTBC, [AMAT], GLBDES, CONST, GRADIENT );  !
1939    5!$                                                                               $!
1940    5!            IF NITER >= OCS AND NITER <= OCE THEN                               !
1941    6!               PRINT("LOG=('         VANGO MODULE')");                          !
1942    6!               CALL VANGO ( NITER, NDV, APPCNVRG, MOVLIM, CNVRGLIM,            !
1943    6!                            CTL, CTLMIN, NUMOPTBC, GLBDES, CONST, [AMAT],      !
1944    6!                            DESHIST );                                         !
1945    6!            ELSE                                                               !
1946    6!               IF NITER >= MPS AND NITER <= MPE THEN                           !
1947    7!                  PRINT("LOG=('         DESIGN MODULE')");                     !
1948    7!                  CALL DESIGN( NITER, NDV, APPCNVRG, MOVLIM, CNVRGLIM,        !
1949    7!                               CTL, CTLMIN, NUMOPTBC, GLBDES, CONST, [AMAT],  !
1950    7!                               DESHIST );                                     !
1951    7!               ENDIF;                                                          !
1952    6!            ENDIF;                                                             !
1953    5!$                                                                              $!
1954    5!         ENDIF; $   END IF ON FSD METHOD                                      $!
1955    4!      ENDIF;    $   END IF TEST AFTER ACTCON                                   $!
1956    3!   ENDDO;       $   END WHILE LOOP FOR GLOBAL CONVERGENCE                      $!
1957    2!ENDIF;          $   END IF ON OPTIMIZATION                                     $!
1958    1!$                                                                              $!
1959    1!$*********************************************************************************$!
1960    1!$        BEGIN FINAL ANALYSIS LOOP                                             $!
1961    1!$*********************************************************************************$!
1962    1!$                                                                              $!
1963    1!IF NBNDCOND > NUMOPTBC THEN                                                    !
1964    2!$                                                                              $!
1965    2!$  ASSEMBLE THE GLOBAL MATRICES                                               $!
1966    2!$                                                                              $!
1967    2!   PRINT("LOG=('******************************')");                            !
1968    2!$                                                                              $!
1969    2!$  ASSEMBLE THE GLOBAL MATRICES                                               $!
1970    2!$  BEGIN BOUNDARY CONDITION LOOP                                              $!
1971    2!$                                                                              $!
1972    2!   PRINT("LOG=('BEGIN FINAL ANALYSIS')");                                      !
1973    2!   CALL ANALINIT;                                                              !
1974    2!   CALL EMA2 ( , NDV, GSIZEB, GLBDES, GMKCT, DKVI, [K1GG],                     !
1975    2!                              GMMCT, DMVI, [M1GG] );                           !
1976    2!   FOR BC = NUMOPTBC + 1 TO NBNDCOND DO                                        !
1977    3!      PRINT("LOG=('      BOUNDARY CONDITION ',I3)",BC);                        !
1978    3!$                                                                             $!
1979    3!$     ESTABLISH THE BASE USET AND PARTITIONING DATA FOR THE BC                 $!
1980    3!$                                                                             $!
1981    3!      CALL MKUSET( BC, GSIZEB, [YS(BC)], [TMN(BC)], [PGMN(BC)], [PNSF(BC)],   !
1982    3!                   [PFOA(BC)], [PARL(BC)], USET(BC) );                        !
1983    3!$                                                                             $!
1984    3!$     MAKE B.C.-DEPENDENT BGPDT FROM BASE, ADDING THE EXTRA POINTS FOR         $!
1985    3!$     THIS B.C.                                                               $!
1986    3!$                                                                             $!
1987    3!      CALL BCBGPDT( BC , GSIZEB , BGPDT(BC) , ESIZE(BC) );                    !
1988    3!      GSIZE     := GSIZEB;                                                    !
1989    3!      PSIZE(BC) := ESIZE(BC) + GSIZE;                                         !
1990    3!$                                                                             $!
1991    3!$     PROCESS MATRICES, TRANSFER FUNCTIONS, AND INITIAL CONDITIONS FOR         $!
1992    3!$     THIS B.C.                                                               $!
1993    3!$                                                                             $!
1994    3!      CALL BCBULK( BC , PSIZE(BC) , BGPDT(BC) , USET(BC) );                   !
1995    3!$                                                                             $!
1996    3!      CALL BOUND  ( BC, GSIZE, ESIZE(BC), USET(BC), BLOAD, BMASS, DMODES,     !
1997    3!                    BMODES, BSAERO, BFLUTR, BDYN, BDRSP, BDTR, BMTR, BDFR,    !
1998    3!                    BMFR, BGUST, BBLAST, NMPC, NSPC, NOMIT, NRSET, NGDR );    !
1999    3!$                                                                             $!
2000    3!$     DETERMINE IF ANY M2GG/K2GG INPUT DATA ARE TO BE ADDED                    $!
2001    3!$                                                                             $!
2002    3!      CALL NULLMAT ( [KGG], [MGG] );                                          !
2003    3!      CALL MK2GG ( BC, GSIZEB, [M2GG], M2GGFLAG, [K2GG], K2GGFLAG );          !
2004    3!      IF M2GGFLAG THEN                                                        !
2005    4!         [MGG] := [M1GG] + [M2GG];                                            !
2006    4!      ELSE                                                                    !
2007    4!         [MGG] := [M1GG];                                                     !
2008    4!      ENDIF;                                                                  !
```

```
2009    3!      IF K2GGFLAG THEN                                                        !
2010    4!         [KGG] := [K1GG] + [K2GG];                                            !
2011    4!      ELSE                                                                    !
2012    4!         [KGG] := [K1GG];                                                     !
2013    4!      ENDIF;                                                                  !
2014    3!$                                                                            $!
2015    3!$     CALL THE GRID POINT WEIGHT GENERATOR FOR THIS BOUNDARY CONDITON        $!
2016    3!$                                                                            $!
2017    3!      CALL GPWG ( , BC, GPWGGRID, [MGG], OGPWG );                            !
2018    3!$                                                                            $!
2019    3!      IF BLOAD <> 0 CALL GTLOAD ( , BC, GSIZE, BGPDT(BC), GLBDES,            !
2020    4!                                 SMPLOD, [DPTHVI], [DPGRVI], [PG], OGRIDLOD); !
2021    3!$                                                                            $!
2022    3!$     PARTITION-REDUCTION OF GLOBAL MATRICES                                 $!
2023    3!$                                                                            $!
2024    3!$************************** TAKEN OUT FOR ZAERO ***************************$! <-
2025    3!$     IF NBNDCOND > 1 CALL NULLMAT ( [KNN], [PN], [MNN], [GTKN], [GSTKN],    $!
2026    3!$                                    [UGTKN] );                              $!
2027    3!$*******************************************************************************$!
2028    3!      IF NBNDCOND > 1 CALL NULLMAT ( [KNN], [PN], [MNN], [UGTKN] );          ! <-
2029    3!      IF NMPC <> 0 THEN                                                       !
2030    4!$                                                                            $!
2031    4!$        PERFORM MPC REDUCTION                                               $!
2032    4!$                                                                            !
2033    4!         PRINT("LOG=('          MPC REDUCTION')");                           !
2034    4!         CALL GREDUCE ( [KGG], [PG], [PGMN(BC)], [TMN(BC)], [KNN], [PN] );   !
2035    4!         IF BMASS <> 0 CALL GREDUCE ([MGG],,[PGMN(BC)], [TMN(BC)], [MNN]);   !
2036    4!$************************** TAKEN OUT FOR ZAERO ***************************$! <-
2037    4!$        IF BSAERO <> 0 THEN                                                 $!
2038    4!$           CALL GREDUCE (, [GTKG], [PGMN(BC)],[TMN(BC)],, [GTKN]);          $!
2039    4!$           CALL GREDUCE (, [GSTKG], [PGMN(BC)],[TMN(BC)],, [GSTKN]);        $!
2040    4!$        ENDIF;                                                              $!
2041    4!$*******************************************************************************$!
2042    4!         IF BFLUTR <> 0 OR BGUST <> 0 OR BBLAST <> 0 OR BSAERO <> 0          ! <-
2043    5!            CALL GREDUCE (, [UGTKG], [PGMN(BC)],[TMN(BC)], , [UGTKN] );     !
2044    4!      ELSE                                                                    !
2045    4!$                                                                            $!
2046    4!$        NO MPC REDUCTION                                                    $!
2047    4!$                                                                            $!
2048    4!         [KNN] := [KGG];                                                     !
2049    4!         IF BLOAD <> 0 [PN] := [PG];                                         !
2050    4!         IF BMASS <> 0 [MNN] := [MGG];                                       !
2051    4!$************************** TAKEN OUT FOR ZAERO ***************************$! <-
2052    4!$        IF BSAERO <> 0 THEN                                                 $!
2053    4!$           [GTKN]  := [GTKG];                                               $!
2054    4!$           [GSTKN] := [GSTKG];                                              $!
2055    4!$        ENDIF;                                                              $!
2056    4!$*******************************************************************************$!
2057    4!         IF BFLUTR <> 0 OR BGUST <> 0 OR BBLAST <> 0 OR BSAERO <> 0          !
2058    5!            [UGTKN]  := [UGTKG];                                             ! <-
2059    4!      ENDIF;                                                                  !
2060    3!$                                                                            $!
2061    3!$     PERFORM AUTOSPC CALCULATIONS ON THE KNN MATRIX                         $! <-
2062    3!$                                                                            $!
2063    3!      PRINT("LOG=('          AUTOSPC COMPUTATIONS')");                       !
2064    3!      CALL GPSP ( , BC, NGDR, [KNN], BGPDT(BC), [YS(BC)], USET(BC),          !
2065    3!                  GPST(BC) );                                                !
2066    3!      CALL MKPVECT ( USET(BC), [PGMN(BC)], [PNSF(BC)], [PFOA(BC)], [PARL(BC)] );!
2067    3!      CALL BOUNDUPD ( BC, GSIZE, ESIZE(BC), USET(BC), NSPC, NOMIT, NRSET );  !
2068    3!$                                                                            $!
2069    3!$************************** TAKEN OUT FOR ZAERO ***************************$! <-
2070    3!$     IF NBNDCOND > 1 CALL NULLMAT ( [KFF], [PF], [MFF], [GTKF], [GSTKF],    $!
2071    3!$                                    [UGTKF] );                              $!
2072    3!$*******************************************************************************$!
2073    3!      IF NBNDCOND > 1 CALL NULLMAT ( [KFF], [PF], [MFF], [UGTKF] );          ! <-
2074    3!      IF NSPC <> 0 THEN                                                       !
2075    4!$                                                                            $!
2076    4!$        PERFORM SPC REDUCTION                                               $!
2077    4!$                                                                            $!
2078    4!         PRINT("LOG=('          SPC REDUCTION')");                           !
2079    4!         CALL NREDUCE ( [KNN], [PN], [PNSF(BC)], [YS(BC)], [KFF], [KFS],     !
2080    4!                        [KSS], [PF], [PS] );                                !
2081    4!         IF BMASS <> 0 CALL NREDUCE ( [MNN], , [PNSF(BC)], , [MFF]);         !
2082    4!$************************** TAKEN OUT FOR ZAERO ***************************$! <-
2083    4!$        IF BSAERO <> 0 THEN                                                 $!
2084    4!$           CALL NREDUCE ( , [GTKN], [PNSF(BC)], , , , , [GTKF] );          $!
2085    4!$           CALL NREDUCE ( , [GSTKN], [PNSF(BC)], , , , , [GSTKF] );        $!
2086    4!$        ENDIF;                                                              $!
2087    4!$*******************************************************************************$!
2088    4!         IF BFLUTR <> 0 OR BGUST <> 0 OR BBLAST <> 0 OR BSAERO <> 0          ! <-
2089    5!            CALL NREDUCE (, [UGTKN], [PNSF(BC)],,,,, [UGTKF]);              !
```

```
2090    4!      ELSE                                                                    !
2091    4!$                                                                           $!
2092    4!$        NO SPC REDUCTION                                                   $!
2093    4!$                                                                           $!
2094    4!         [KFF] := [KNN];                                                     !
2095    4!         IF BLOAD <> 0 [PF] := [PN];                                         !
2096    4!         IF BMASS <> 0 [MFF] := [MNN];                                       !
2097    4!$*************************** TAKEN OUT FOR ZAERO ****************************$!
2098    4!$        IF BSAERO <> 0 THEN                                                $!
2099    4!$           [GTKF]  := [GTKN];                                              $!
2100    4!$           [GSTKF] := [GSTKN];                                             $!
2101    4!$        ENDIF;                                                             $!
2102    4!$*************************************************************************$!
2103    4!         IF BFLUTR <> 0 OR BGUST <> 0 OR BBLAST <> 0 OR BSAERO <> 0          !
2104    5!            [UGTKF]  := [UGTKN];                                             !
2105    4!      ENDIF;                                                                 !
2106    3!$                                                                           $!
2107    3!      IF NBNDCOND > 1 CALL NULLMAT ([KAA], [PA], [MAA], [KAAA], [PAA], [UGTKA]);!
2108    3!$                                                                           $!
2109    3!      IF NGDR <> 0 THEN                                                      !
2110    4!$                                                                          $!
2111    4!$        PERFORM THE GENERAL DYNAMIC REDUCTION WHICH IS DISCIPLINE          $!
2112    4!$        INDEPENDENT.  THE RESULTING [GSUBO] MATRIX WILL BE USED BY        $!
2113    4!$        ALL DISCIPLINES                                                    $!
2114    4!$                                                                          $!
2115    4!         PRINT("LOG=('          DYNAMIC REDUCTION')");                       !
2116    4!$                                                                          $!
2117    4!$        OBTAIN THE OMITTED DOF PARTITION OF KFF AND MFF                    $!
2118    4!$                                                                          $!
2119    4!         CALL PARTN ( [KFF], [KOO], , [KOA], , [PFOA(BC)] );                 !
2120    4!         CALL PARTN ( [MFF], [MOO], , , , [PFOA(BC)] );                      !
2121    4!         ASIZE := GSIZE - NMPC - NSPC - NOMIT;                               !
2122    4!         LSIZE := ASIZE - NRSET;                                             !
2123    4!         CALL GDR1 ( [KOO], [MOO], [KSOO], [GGO], LKSET, LJSET, NEIV,       !
2124    4!                     FMAX, BC, BGPDT(BC), USET(BC), NOMIT, LSIZE );          !
2125    4!$                                                                          $!
2126    4!$        LKSET              MEANING                                         $!
2127    4!$              <> 0         APPROX. MODE SHAPES SELECTED                    $!
2128    4!$               = 0         NO APPROX. MODE SHAPES IN GDR                   $!
2129    4!$                                                                          $!
2130    4!         IF LKSET <> 0  THEN                                                 !
2131    5!            CALL SDCOMP ( [KSOO], [LSOO], USET(BC), SINGOSET );              !
2132    5!            CALL GDR2   ( [LSOO], [MOO], [PHIOK], LKSET, LJSET,              !
2133    5!                          NEIV,   FMAX,  BC );                               !
2134    5!         ENDIF;                                                              !
2135    4!         CALL GDR3 ( [KOO], [KOA], [MGG], [PHIOK], [TMN(BC)], [GGO],        !
2136    4!                     [PGMN(BC)], [PNSF(BC)], [PFOA(BC)], [GSUBO(BC)],        !
2137    4!                     BGPDT(BC), USET(BC),                                    !
2138    4!                     LKSET, LJSET, ASIZE, GNORM, BC );                       !
2139    4!         CALL GDR4 ( BC, GSIZE, PSIZE(BC), LKSET, LJSET, NUMOPTBC, NBNDCOND, !
2140    4!                     [PGMN(BC)], [TMN(BC)], [PNSF(BC)], [PFOA(BC)],          !
2141    4!                     [PARL(BC)], [PGDRG(BC)], [PAJK], [PFJK], BGPDT(BC),     !
2142    4!                     USET(BC) );                                             !
2143    4!      ENDIF;                                                                 !
2144    3!$                                                                          $!
2145    3!      IF BLOAD <> 0 OR BMODES <> 0 OR BFLUTR <> 0 OR BDYN <> 0 THEN        !
2146    4!$                                                                          $!
2147    4!$        REDUCE THE MATRICES WITHOUT AEROELASTIC CORRECTIONS                $!
2148    4!$                                                                          $!
2149    4!         IF NGDR <> 0 THEN                                                   !
2150    5!$                                                                          $!
2151    5!$           PERFORM THE GENERAL DYNAMIC REDUCTION                          $!
2152    5!$                                                                          $!
2153    5!            PRINT("LOG=('          SYMMETRIC DYNAMIC REDUCTION')");         !
2154    5!$                                                                          $!
2155    5!            [MAA] := TRANS ( [GSUBO(BC)] ) * [ [MFF] * [GSUBO(BC)] ];       !
2156    5!            [KAA] := TRANS ( [GSUBO(BC)] ) * [ [KFF] * [GSUBO(BC)] ];       !
2157    5!            IF BLOAD <> 0  [PA]    := TRANS ( [GSUBO(BC)] ) * [PF];         !
2158    5!            IF BFLUTR <> 0 OR BGUST <> 0 OR BBLAST <> 0 THEN                !
2159    6!               [TMP1]  := TRANS ( [UGTKF] ) * [GSUBO(BC)];                  !
2160    6!               CALL TRNSPOSE ( [TMP1], [UGTKA] );                           !
2161    6!            ENDIF;                                                          !
2162    5!         ELSE                                                               !
2163    5!            IF NOMIT <> 0 THEN                                              !
2164    6!$                                                                          $!
2165    6!$              PERFORM THE STATIC REDUCTION                                $!
2166    6!$                                                                          $!
2167    6!               PRINT("LOG=('          STATIC CONDENSATION')");              !
2168    6!$                                                                          $!
2169    6!               CALL FREDUCE ( [KFF], [PF], [PFOA(BC)], , [KOOINV(BC)], , ,  !
2170    6!                              [GSUBO(BC)], [KAA], [PA], [PO], USET(BC) );   !
```

```
2171    6!$                                                                    $!
2172    6!            IF BMASS <> 0 THEN                                        !
2173    7!$                                                                    $!
2174    7!$              PERFORM GUYAN REDUCTION OF THE MASS MATRIX             $!
2175    7!$                                                                    $!
2176    7!               CALL PARTN ( [MFF], [MOO], , [MOA], [MAABAR], [PFOA(BC)] );  !
2177    7!               [MAA] := [MAABAR] + TRANS([MOA]) * [GSUBO(BC)] +          !
2178    7!                        TRANS([GSUBO(BC)]) * [MOA] +                     !
2179    7!                        TRANS([GSUBO(BC)]) * [ [MOO] * [GSUBO(BC)] ];    !
2180    7!               IF NRSET <> 0 [IFM(BC)] := [MOO] * [GSUBO(BC)] + [MOA];   !
2181    7!            ENDIF;                                                       !
2182    6!            IF BFLUTR <> 0 OR BGUST <> 0 OR BBLAST <> 0 THEN             !
2183    7!               CALL ROWPART ( [UGTKF], [UGTKO], [UGTKAB], [PFOA(BC)] );  !
2184    7!               [TMP1]  := TRANS( [UGTKO] ) * [GSUBO(BC)];                !
2185    7!               CALL TRNSPOSE ( [TMP1], [TMP2] );                         !
2186    7!               [UGTKA] := [UGTKAB] + [TMP2];                             !
2187    7!            ENDIF;                                                       !
2188    6!         ELSE                                                            !
2189    6!$                                                                    $!
2190    6!$              NO F-SET REDUCTION                                    $!
2191    6!$                                                                    $!
2192    6!               [KAA] := [KFF];                                           !
2193    6!               IF BLOAD <> 0 [PA] := [PF];                               !
2194    6!               IF BFLUTR <> 0 OR BGUST <> 0 OR BBLAST <> 0 [UGTKA]:=[UGTKF];  !
2195    6!               IF BMASS <> 0 [MAA] := [MFF];                             !
2196    6!            ENDIF;                                                       !
2197    5!         ENDIF;                                                          !
2198    4!$                                                                    $!
2199    4!         IF NRSET <> 0 THEN                                              !
2200    5!$                                                                    $!
2201    5!$            PERFORM THE SUPPORT SET REDUCTION                       $!
2202    5!$                                                                    $!
2203    5!            PRINT("LOG=('          SUPPORT REDUCTION')");                !
2204    5!            CALL PARTN  ( [KAA], [KRR], [KLR], , [KLL], [PARL(BC)] );   !
2205    5!            CALL SDCOMP ( [KLL], [KLLINV(BC)], USET(BC), SINGLSET );     !
2206    5!            CALL FBS ( [KLLINV(BC)], [KLR], [D(BC)], -1 );               !
2207    5!            CALL RBCHECK ( BC, USET(BC), BGPDT(BC), [D(BC)], [KLL],      !
2208    5!                           [KRR], [KLR] );                               !
2209    5!$                                                                    $!
2210    5!$            CALCULATE THE REDUCED MASS MATRIX                       $!
2211    5!$                                                                    $!
2212    5!            CALL PARTN ([MAA], [MRRBAR], [MLR], , [MLL], [PARL(BC)]);    !
2213    5!            [IFR(BC)] := [MLL] * [D(BC)] + [MLR];                        !
2214    5!            [MRR(BC)] := [MRRBAR] + TRANS ( [MLR] )   * [D(BC)] +        !
2215    5!                                    TRANS ( [D(BC)] ) * [IFR(BC)];       !
2216    5!            [R22]     := TRANS ( [D(BC)] ) * [MLR] + [MRRBAR];           !
2217    5!$                                                                    $!
2218    5!            IF BLOAD <> 0 THEN                                           !
2219    6!$                                                                    $!
2220    6!$               PROCESS STATICS WITH INERTIA RELIEF                  $!
2221    6!$                                                                    $!
2222    6!               PRINT("LOG=('         >>>DISCIPLINE: STATICS(INERTIA RELIEF)')");!
2223    6!               CALL ROWPART ( [PA], [PR], [PLBAR], [PARL(BC)] );         !
2224    6!               [LHS(BC)] := [MRR(BC)];                                   !
2225    6!               [RHS(BC)] := TRANS([D(BC)]) * [PLBAR] + [PR];             !
2226    6!               CALL INERTIA ( [LHS(BC)], [RHS(BC)], [AR] );              !
2227    6!               [AL]      := [D(BC)] * [AR];                              !
2228    6!               CALL ROWMERGE ( [AA], [AR], [AL], [PARL(BC)] );           !
2229    6!               [RHS(BC)] := [PLBAR] - [IFR(BC)] * [AR];                  !
2230    6!               CALL FBS ( [KLLINV(BC)], [RHS(BC)], [UL] );               !
2231    6!               CALL YSMERGE ( [UA], , [UL], [PARL(BC)] );                !
2232    6!            ENDIF;                                                       !
2233    5!            IF BMODES <> 0 THEN                                          !
2234    6!               PRINT("LOG=('         >>>DISCIPLINE: NORMAL MODES')");    !
2235    6!               CALL REIG ( , BC, USET(BC), [KAA], [MAA], [MRR(BC)],      !
2236    6!                           [D(BC)], LAMBDA, [PHIA], [MII], HSIZE(BC) );  !
2237    6!               CALL OFPMROOT ( , BC, NUMOPTBC, LAMBDA );                 !
2238    6!            ENDIF;                                                       !
2239    5!         ELSE                                                            !
2240    5!$                                                                    $!
2241    5!$            NO SUPPORT SET REDUCTION                                $!
2242    5!$                                                                    $!
2243    5!            IF BLOAD <> 0 THEN                                           !
2244    6!               PRINT("LOG=('         >>>DISCIPLINE: STATICS')");         !
2245    6!               CALL SDCOMP ( [KAA], [KLLINV(BC)], USET(BC), SINGASET );  !
2246    6!               CALL FBS ( [KLLINV(BC)], [PA], [UA] );                    !
2247    6!            ENDIF;                                                       !
2248    5!            IF BMODES <> 0 THEN                                          !
2249    6!               PRINT("LOG=('         >>>DISCIPLINE: NORMAL MODES')");    !
2250    6!               CALL REIG ( , BC, USET(BC), [KAA], [MAA], , , LAMBDA,     !
2251    6!                           [PHIA], [MII], HSIZE(BC) );                   !
```

```
2252   6!              CALL OFPMROOT ( , BC, NUMOPTBC, LAMBDA );                    !
2253   6!            ENDIF;                                                         !
2254   5!          ENDIF;                                                           !
2255   4!        ENDIF;                                                           !
2256   3!$                                                                       $!
2257   3!        IF BSAERO <> 0 THEN                                              !
2258   4!$                                                                       $!
2259   4!$         PERFORM STATIC AERODYNAMIC ANALYSES                           $!
2260   4!$                                                                       $!
2261   4!          PRINT("LOG=('          SAERO INITIALIZATION')");               !
2262   4!$************************** TAKEN OUT FOR ZAERO ****************************$! ←
2263   4!$         CALL TRNSPOSE ( [GSTKF], [GSKF] );                            $!
2264   4!$*************************************************************************$!
2265   4!          CALL TRNSPOSE ( [UGTKF], [GSKF] );                            ! ←
2266   4!          LOOP := TRUE;                                                  !
2267   4!          SUB  := 0;                                                     !
2268   4!          WHILE LOOP DO                                                  !
2269   5!            SUB := SUB + 1;                                              !
2270   5!            CALL SAERODRV (BC, SUB, LOOP, MINDEX, SYM, MACH, QDP, 1 );   !
2271   5!$                                                                       $!
2272   5!$           ADJUST THE KFF MATRIX AND DETERMINE THE RIGID AIR LOADS      $!
2273   5!$                                                                       $!
2274   5!$************************** TAKEN OUT FOR ZAERO ****************************$! ←
2275   5!$           IF SYM = 1  [AICS] := [GSTKF]*[TRANS([AICMAT(MINDEX)])*[GSKF]];  $!
2276   5!$           IF SYM = -1 [AICS] := [GSTKF]*[TRANS([AAICMAT(MINDEX)])*[GSKF]]; $!
2277   5!$           [PAF]  := (QDP) [ [GSTKF] * [AIRFRC(MINDEX)] ];              $!
2278   5!$*************************************************************************$!
2279   5!            IF SYM = 1  [AICS] := [UGTKF]*[TRANS([AICMAT(MINDEX)])*[GSKF]];  !
2280   5!            IF SYM = -1 [AICS] := [UGTKF]*[TRANS([AAICMAT(MINDEX)])*[GSKF]]; !
2281   5!            [PAF]  := (QDP) [ [UGTKF] * [AIRFRC(MINDEX)] ];              ! ←
2282   5!            [KAFF] := [KFF] - (QDP) [AICS];                              !
2283   5!$                                                                       $!
2284   5!$           REDUCE THE MATRICES WITH AEROELASTIC CORRECTIONS              $!
2285   5!$           SAVE THE SUBCASE/BC DEPENDENT DATA FOR SENSITIVITY ANALYSIS   $!
2286   5!$                                                                       $!
2287   5!            IF NGDR <> 0 THEN                                            !
2288   6!$                                                                       $!
2289   6!$             PERFORM THE GENERAL DYNAMIC REDUCTION                      $!
2290   6!$                                                                       $!
2291   6!              PRINT("LOG=('          SAERO DYNAMIC REDUCTION')");         !
2292   6!              [MAAA] := TRANS ( [GSUBO(BC)] ) * [ [MFF] * [GSUBO(BC)] ];   !
2293   6!              [KAAA] := TRANS ( [GSUBO(BC)] ) * [ [KAFF] * [GSUBO(BC)] ];  !
2294   6!              [PAA]  := TRANS ( [GSUBO(BC)] ) * [PAF];                     !
2295   6!            ELSE                                                           !
2296   6!              IF NOMIT <> 0 THEN                                           !
2297   7!$                                                                       $!
2298   7!$             PERFORM THE STATIC REDUCTION                               $!
2299   7!$                                                                       $!
2300   7!              PRINT("LOG=('          SAERO STATIC CONDENSATION')");       !
2301   7!$                                                                       $!
2302   7!              IF NRSET <> 0 AND SUB = 1 AND BLOAD = 0 AND BMODES = 0 AND   !
2303   8!                 BFLUTR = 0 AND BDYN = 0 THEN                           !
2304   8!$                                                                       $!
2305   8!$             FORM [KAA] ON SO [D] CAN BE FORMED                         $!
2306   8!$                                                                       $!
2307   8!              CALL FREDUCE ([KFF], , [PFOA(BC)], , [KOOINV(BC)], , ,       !
2308   8!                           [GSUBO(BC)], [KAA], , , USET(BC) );             !
2309   8!              ENDIF;                                                       !
2310   7!$                                                                       $!
2311   7!              CALL FREDUCE ( [KAFF], [PAF], [PFOA(BC)], BSAERO,            !
2312   7!                             [KOOL(BC,SUB)], [KOOU(BC,SUB)],               !
2313   7!                             [KAO(BC,SUB)], [GASUBO(BC,SUB)], [KAAA],      !
2314   7!                             [PAA], [POARO(BC,SUB)], USET(BC));            !
2315   7!$                                                                       $!
2316   7!              IF BMASS <> 0 THEN                                           !
2317   8!$                                                                       $!
2318   8!$             PERFORM GUYAN REDUCTION OF THE MASS MATRIX                 $!
2319   8!$                                                                       $!
2320   8!              CALL PARTN ( [MFF], [MOO], , [MOA], [MAABAR],                !
2321   8!                           [PFOA(BC)] );                                   !
2322   8!              [MAAA] := [MAABAR] + TRANS([MOA]) * [GASUBO(BC,SUB)] +       !
2323   8!                        TRANS([GASUBO(BC,SUB)]) * [MOA] +                  !
2324   8!                        TRANS([GASUBO(BC,SUB)]) * [[MOO] *                 !
2325   8!                        [GASUBO(BC,SUB)]];                                 !
2326   8!              IF NRSET <> 0                                                !
2327   9!                        [IFMA(BC,SUB)] := [MOO]*[GASUBO(BC,SUB)]+[MOA];    !
2328   8!              ENDIF;                                                       !
2329   7!            ELSE                                                           !
2330   7!$                                                                       $!
2331   7!$             NO F-SET REDUCTION                                         $!
2332   7!$                                                                       $!
```

```
2333    7!              IF NRSET <> 0 AND SUB    = 1 AND BLOAD = 0  AND                !
2334    8!                 BMODES = 0 AND BFLUTR = 0 AND BDYN = 0 THEN                 !
2335    8!$                                                                           $!
2336    8!$                 FORM [KAA] ON FIRST PASS SO [D] CAN BE FORMED              $!
2337    8!$                                                                           $!
2338    8!                  [KAA] := [KFF];                                            !
2339    8!               ENDIF;                                                        !
2340    7!               [KAAA] := [KAFF];                                             !
2341    7!               [MAAA] := [MFF];                                              !
2342    7!               [PAA]  := [PAF];                                              !
2343    7!            ENDIF;                                                           !
2344    6!         ENDIF;                                                             !
2345    5!$                                                                           $!
2346    5!         IF NRSET <> 0 THEN                                                 !
2347    6!$                                                                           $!
2348    6!$            PERFORM THE SUPPORT SET REDUCTION                              $!
2349    6!$                                                                           $!
2350    6!             PRINT("LOG=('          SAERO SUPPORT REDUCTION')");             !
2351    6!$                                                                           $!
2352    6!             IF SUB = 1 AND BLOAD = 0 AND BMODES = 0 AND BFLUTR = 0         !
2353    7!                        AND BDYN  = 0 THEN                                  !
2354    7!$                                                                          $!
2355    7!$               [D] WAS NOT COMPUTED FOR NON-SAERO DISCIPLINES SO           $!
2356    7!$               NEED TO COMPUTE IT NOW                                      $!
2357    7!$                                                                           $!
2358    7!               CALL PARTN  ( [KAA], [KRR], [KLR], , [KLL], [PARL(BC)] );   !
2359    7!               CALL SDCOMP ( [KLL], [KLLINV(BC)], USET(BC), SINGLSET );     !
2360    7!               CALL FBS ( [KLLINV(BC)], [KLR], [D(BC)], -1 );               !
2361    7!               CALL RBCHECK ( BC, USET(BC), BGPDT(BC), [D(BC)], [KLL],      !
2362    7!                              [KRR], [KLR] );                               !
2363    7!             ENDIF;                                                         !
2364    6!$                                                                          $!
2365    6!$            CALCULATE THE REDUCED MASS MATRIX                             $!
2366    6!$                                                                          $!
2367    6!             CALL PARTN ([MAAA], [MRRBAR], [MLR], , [MLL], [PARL(BC)]);   !
2368    6!             [R13(BC,SUB)] := [MLL] * [D(BC)] + [MLR];                     !
2369    6!             [R33]         := [MRRBAR] + TRANS ( [MLR] ) * [D(BC)] +       !
2370    6!                              TRANS ( [D(BC)] ) * [R13(BC,SUB)];           !
2371    6!             [R22]         := TRANS ( [D(BC)] ) * [MLR] + [MRRBAR];        !
2372    6!             CALL TRNSPOSE ( [R13(BC,SUB)], [R21(BC,SUB)] );               !
2373    6!$                                                                          $!
2374    6!$            PROCESS STEADY AEROELASTIC DISCIPLINE                         $!
2375    6!$                                                                          $!
2376    6!             PRINT("LOG=('          >>>DISCIPLINE: STEADY AERO')");        !
2377    6!             CALL PARTN ( [KAAA], [KARR], [R12(BC,SUB)], [KARL], [R11],   !
2378    6!                          [PARL(BC)] );                                    !
2379    6!             [R32(BC,SUB)] := TRANS([D(BC)]) * [R12(BC,SUB)] + [KARR];     !
2380    6!             [R31(BC,SUB)] := TRANS([D(BC)]) * [R11] + [KARL];             !
2381    6!$                                                                          $!
2382    6!             CALL DECOMP ( [R11], [RL11(BC,SUB)], [RU11(BC,SUB)] );       !
2383    6!$                                                                          $!
2384    6!             CALL ROWPART ( [PAA], [PARBAR], [PAL], [PARL(BC)] );          !
2385    6!             CALL GFBS ( [RL11(BC,SUB)], [RU11(BC,SUB)], [PAL],            !
2386    6!                         [R11PAL(BC,SUB)], -1);                           !
2387    6!             [PRIGID] := [PARBAR] + TRANS([D(BC)]) * [PAL];                !
2388    6!             [P1]     := [R21(BC,SUB)] * [R11PAL(BC,SUB)];                !
2389    6!             [P2]     := [PRIGID] + [R31(BC,SUB)] * [R11PAL(BC,SUB)];      !
2390    6!$                                                                          $!
2391    6!             CALL GFBS ( [RL11(BC,SUB)], [RU11(BC,SUB)], [R12(BC,SUB)],   !
2392    6!                         [R1112(BC,SUB)], -1);                            !
2393    6!             CALL GFBS ( [RL11(BC,SUB)], [RU11(BC,SUB)], [R13(BC,SUB)],   !
2394    6!                         [R1113(BC,SUB)], -1);                            !
2395    6!             [K11]         := [R22] + [R21(BC,SUB)] * [R1112(BC,SUB)];     !
2396    6!             [K12(BC,SUB)] := [R21(BC,SUB)] * [R1113(BC,SUB)];             !
2397    6!             [K21(BC,SUB)] := [R32(BC,SUB)] + [R31(BC,SUB)] * [R1112(BC,SUB)];!
2398    6!             [K22]         := [R33] + [R31(BC,SUB)] * [R1113(BC,SUB)];     !
2399    6!$                                                                          $!
2400    6!             CALL DECOMP ( [K11], [KL11(BC,SUB)], [KU11(BC,SUB)] );       !
2401    6!             CALL GFBS ( [KL11(BC,SUB)], [KU11(BC,SUB)], [P1],            !
2402    6!                         [PAR(BC,SUB)] );                                 !
2403    6!             CALL GFBS ( [KL11(BC,SUB)], [KU11(BC,SUB)], [K12(BC,SUB)],   !
2404    6!                         [K1112(BC,SUB)],-1 );                            !
2405    6!             [LHSA(BC,SUB)] := [K22] + [K21(BC,SUB)] * [K1112(BC,SUB)];   !
2406    6!             [RHSA(BC,SUB)] := [P2] - [K21(BC,SUB)] * [PAR(BC,SUB)];      !
2407    6!             CALL SAERO ( , BC, MINDEX, SUB, SYM, QDP, STABCF,             !
2408    6!                          BGPDT(BC), [LHSA(BC,SUB)], [RHSA(BC,SUB)], [AAR],!
2409    6!                          [DELTA(SUB)], [PRIGID], [R33] );                 !
2410    6!             [AAL]         := [D(BC)] * [AAR];                            !
2411    6!             CALL ROWMERGE ( [AAA(SUB)], [AAR], [AAL], [PARL(BC)] );      !
2412    6!             [UAR]    := [K1112(BC,SUB)] * [AAR] + [PAR(BC,SUB)] *        !
2413    6!                         [DELTA(SUB)];                                    !
```

```
2414   6!              [UAL]    := [R1112(BC,SUB)] * [UAR] + [R1113(BC,SUB)] * [AAR]      !
2415   6!                        - [R11PAL(BC,SUB)] * [DELTA(SUB)];                        !
2416   6!              CALL ROWMERGE ( [UAA(SUB)], [UAR], [UAL], [PARL(BC)] );           !
2417   6!              IF NOMIT <> 0 [PAO(SUB)] := [POARO(BC,SUB)] * [DELTA(SUB)] ;      !
2418   6!           ELSE                                                                 !
2419   6!$                                                                             $!
2420   6!$             NO SUPPORT SET REDUCTION                                        $!
2421   6!$                                                                             $!
2422   6!$                                                                             $!
2423   6!$             PROCESS STEADY AEROELASTIC DISCIPLINE                           $!
2424   6!$                                                                             $!
2425   6!              PRINT("LOG=('          >>>DISCIPLINE: STEADY AERO')");           !
2426   6!           ENDIF;                                                              !
2427   5!        ENDDO;                                                                 !
2428   4!     ENDIF;                                                                    !
2429   3!$                                                                            $!
2430   3!$    PERFORM ANY DYNAMIC ANALYSES -- NOTE THAT THESE ARE INDEPENDENT          $!
2431   3!$    OF THE SUPPORT SET                                                       $!
2432   3!$                                                                            $!
2433   3!     IF BDYN <> 0 THEN                                                        !
2434   4!        IF BFLUTR <> 0 THEN                                                   !
2435   5!           PRINT("LOG=('          >>>DISCIPLINE: FLUTTER')");                 !
2436   5!           SUB  := 0;                                                        !
2437   5!           LOOP := TRUE;                                                     !
2438   5!           WHILE LOOP DO                                                     !
2439   6!              SUB := SUB + 1;                                                !
2440   6!              CALL FLUTDRV ( BC, SUB, LOOP );                                !
2441   6!              CALL FLUTQHHZ ( , BC, SUB, ESIZE(BC), PSIZE(BC), [AJK],        ! ←
2442   6!                              [SKJ], [UGTKA], [PHIA], USET(BC),              !
2443   6!                              [TMN(BC)], [GSUBO(BC)], NGDR, AECOMPU, GEOMUA, !
2444   6!                              [PHIKH], [QHHLFL(BC,SUB)], OAGRDDSP );         ! ←
2445   6!              CALL FLUTDMA ( , BC, SUB, ESIZE(BC), PSIZE(BC), BGPDT(BC),     !
2446   6!                             USET(BC), [MAA], [KAA], [TMN(BC)],[GSUBO(BC)],  !
2447   6!                             NGDR, LAMBDA, [PHIA], [MHHFL(BC,SUB)],          !
2448   6!                             [BHHFL(BC,SUB)], [KHHFL(BC,SUB)] );             !
2449   6!              CALL FLUTTRAZ ( , BC, SUB, [QHHLFL(BC,SUB)], LAMBDA, KSIZE(BC), ! ←
2450   6!                              ESIZE(BC), [MKHFL(BC,SUB)], [BHHFL(BC,SUB)],   !
2451   6!                              [KHHFL(BC,SUB)], CLAMBDA, ,AEROZ );            ! ←
2452   6!           ENDDO;                                                            !
2453   5!        ENDIF;                                                               !
2454   4!        IF BDRSP <> 0 THEN                                                   !
2455   5!           IF BMTR <> 0 OR BDTR <> 0 THEN                                    !
2456   6!              PRINT("LOG=('          >>>DISCIPLINE: TRANSIENT RESPONSE')");  !
2457   6!           ENDIF;                                                            !
2458   5!           IF BMFR <> 0 OR BDFR <> 0 THEN                                    !
2459   6!              PRINT("LOG=('          >>>DISCIPLINE: FREQUENCY RESPONSE')");  !
2460   6!           ENDIF;                                                            !
2461   5!$$$$$$$$$$$$$$$$$$$$$$ MODIFIED FOR ZAERO $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$! ←
2462   5!$           CALL QHHLGEN (BC, ESIZE(BC), [QKKL], [QKJL], [UGTKA], [PHIA],    $!
2463   5!$                         [PHIKH], [QHHL], [QHJL]);                         $!
2464   5!$$$$$$$$$$$$$$$$$$$$$$ MODIFIED FOR ZAERO $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$!
2465   5!           CALL QHHLGENZ (BC, ESIZE(BC),[AJK],[SKJ],[QGK],[UGTKA], [PHIA],  !
2466   5!                          [PHIKH], [QHHL], [QHJL], AEROZ);                   ! ←
2467   5!           CALL DMA ( , BC, ESIZE(BC), PSIZE(BC), BGPDT(BC), USET(BC), [MAA], !
2468   5!                     [KAA], [TMN(BC)], [GSUBO(BC)], NGDR,                    !
2469   5!                     LAMBDA, [PHIA], [MDD], [BDD], [KDDT], [KDDF],           !
2470   5!                     [MHH], [BHH], [KHHT], [KHHF] );                         !
2471   5!           CALL DYNLOAD ( , BC, GSIZE, ESIZE(BC), PSIZE(BC), SMPLOD, BGPDT(BC),!
2472   5!                          USET(BC), [TMN(BC)], [GSUBO(BC)],                  !
2473   5!                          NGDR, [PHIA], [QHJL], [PDT], [PDF],                !
2474   5!                          [PTGLOAD], [PTHLOAD], [PFGLOAD], [PFHLOAD] );      !
2475   5!           CALL DYNRSP (BC, ESIZE(BC), [MDD], [BDD], [KDDT], [KDDF],       !
2476   5!                        [MHH], [BHH], [KHHT], [KHHF], [PDT], [PDF],        !
2477   5!                        [QHHL], [UTRANA], [UFREQA], [UTRANI], [UFREQI],     !
2478   5!                        [UTRANE], [UFREQE] );                               !
2479   5!           IF BMTR <> 0 [UTRANA] := [PHIA] * [UTRANI];                       !
2480   5!           IF BMFR <> 0 [UFREQA] := [PHIA] * [UFREQI];                       !
2481   5!        ENDIF;                                                               !
2482   4!     ENDIF;                                                                  !
2483   3!     IF BBLAST <> 0 THEN                                                     !
2484   4!        PRINT("LOG=('          >>>DISCIPLINE: BLAST')");                     !
2485   4!        CALL BLASTFIT ( BC, [QJJL], [MATTR], [MATSS], BQDP, [BFRC],         !
2486   4!                        [DWNWSH], HSIZE(BC), [ID2], [MPART], [UGTKA],       !
2487   4!                        [BLGTJA], [BLSTJA] );                               !
2488   4!        CALL COLPART ( [PHIA], , [PHIE], [MPART] );                         !
2489   4!        CALL ROWMERGE ( [PHIR], [ID2], [D(BC)], [PARL(BC)] );               !
2490   4!        CALL COLMERGE ( [PHIB], [PHIR], [PHIE], [MPART] );                  !
2491   4!        [GENM]   := TRANS( [PHIB] ) * [ [MAA] * [PHIB] ];                   !
2492   4!        [GENK]   := TRANS( [PHIB] ) * [ [KAA] * [PHIB] ];                   !
2493   4!        [DTSLP]  := TRANS ( [BLSTJA] ) * [PHIB];                            !
2494   4!        [FTF]    := TRANS ( [PHIB] ) * [BLGTJA];                            !
```

```
2495    4!          [GENF]    := (BQDP) [FTF] * [BFRC];                                    !
2496    4!          [GENFA]   := (BQDP) [FTF] * [MATSS];                                   !
2497    4!          [GENQ]    := [GENFA] * [DTSLP];                                        !
2498    4!          [GENQL]   := (BQDP) [FTF] * [MATTR];                                   !
2499    4!          CALL PARTN ( [GENQ],[QRR] , , [QRE], [QEE], [MPART] );                 !
2500    4!          CALL PARTN ( [GENK], , , , [KEE], [MPART] );                           !
2501    4!          [KEQE]    := [QEE] + [KEE];                                            !
2502    4!          CALL DECOMP ( [KEQE], [LKQ], [UKQ] );                                 !
2503    4!          CALL ROWPART ( [GENF], [GFR], [GFE], [MPART] );                        !
2504    4!          CALL GFBS ( [LKQ], [UKQ], [GFE], [BTEM] );                            !
2505    4!          [DELM]    := -[QRE] * [BTEM] + [GFR];                                  !
2506    4!          CALL BLASTRIM ( BC, [DELM], [MRR(BC)], [URDB], [DELB] );               !
2507    4!          [ELAS]    := [BTEM] * [DELB];                                          !
2508    4!          [SLPMOD] := TRANS ( [BLSTJA] ) * [PHIE];                               !
2509    4!          CALL BLASTDRV ( BC, [GENM], [GENK], [GENFA], [GENQL], [DELB],          !
2510    4!                          [URDB], [DWNWSH], [SLPMOD], [ELAS], [UBLASTI] );       !
2511    4!       ENDIF;                                                                    !
2512    3!$                                                                              $!
2513    3!$    BEGIN THE DATA RECOVERY OPERATIONS                                        $!
2514    3!$                                                                              $!
2515    3!     IF NBNDCOND > 1 CALL NULLMAT ( [UF], [AF], [PHIF] );                      !
2516    3!     IF NGDR <> 0 THEN                                                         !
2517    4!$                                                                             $!
2518    4!$       DATA RECOVERY WITH GDR                                                $!
2519    4!$       APPEND THE GDR-GENERATED DOFS TO THE F-SET                            $!
2520    4!$                                                                             $!
2521    4!        PRINT("LOG=('          DYNAMIC REDUCTION RECOVERY')");                 !
2522    4!        IF BLOAD <> 0 THEN                                                     !
2523    5!           [UFGDR] := [GSUBO(BC)] * [UA];                                      !
2524    5!           CALL ROWPART ( [UA], [UJK], , [PAJK] );                             !
2525    5!           CALL ROWMERGE ( [UF], [UJK], [UFGDR], [PFJK] );                     !
2526    5!           IF NRSET <> 0 THEN                                                  !
2527    6!              [AFGDR] := [GSUBO(BC)] * [AA];                                   !
2528    6!              CALL ROWPART ( [AA], [UJK], , [PAJK] );                          !
2529    6!              CALL ROWMERGE ( [AF], [UJK], [AFGDR], [PFJK] );                  !
2530    6!           ENDIF;                                                              !
2531    5!        ENDIF;                                                                 !
2532    4!        IF BSAERO <> 0 THEN                                                    !
2533    5!           FOR S = 1 TO SUB DO                                                 !
2534    6!              [UFGDR] := [GSUBO(BC)] * [UAA(S)];                               !
2535    6!              CALL ROWPART ( [UAA(S)], [UJK], , [PAJK] );                      !
2536    6!              CALL ROWMERGE ( [UAFTMP], [UJK], [UFGDR], [PFJK] );             !
2537    6!$                                                                             $!
2538    6!$              MERGE THE CURRENT SUBCASE DEPENDENT RESULTS INTO A SINGLE       $!
2539    6!$              MATRIX OF RESPONSE QUANTITIES FOR FURTHER RECOVERY              $!
2540    6!$                                                                             $!
2541    6!              CALL SAEROMRG ( BC, S, [UAF], [UAFTMP] );                       !
2542    6!              IF NRSET <> 0 THEN                                              !
2543    7!                 [AFGDR] := [GSUBO(BC)] * [AAA(S)];                           !
2544    7!                 CALL ROWPART ( [AAA(S)], [UJK], , [PAJK] );                  !
2545    7!                 CALL ROWMERGE ( [AAFTMP], [UJK], [AFGDR], [PFJK] );          !
2546    7!                 CALL SAEROMRG ( BC, S, [AAF], [AAFTMP] );                    !
2547    7!              ENDIF;                                                          !
2548    6!           ENDDO;                                                             !
2549    5!        ENDIF;                                                                 !
2550    4!        IF BMODES <> 0 THEN                                                    !
2551    5!           [UFGDR]   := [GSUBO(BC)] * [PHIA];                                  !
2552    5!           CALL ROWPART ( [PHIA], [UJK], , [PAJK] );                          !
2553    5!           CALL ROWMERGE ( [PHIF], [UJK], [UFGDR], [PFJK] );                  !
2554    5!        ENDIF;                                                                 !
2555    4!        IF BDTR   <> 0 OR BMTR <> 0 THEN                                       !
2556    5!           [UFGDR] := [GSUBO(BC)] * [UTRANA];                                  !
2557    5!           CALL ROWPART ( [UTRANA], [UJK], , [PAJK] );                         !
2558    5!           CALL ROWMERGE ( [UTRANF], [UJK], [UFGDR], [PFJK] );                 !
2559    5!        ENDIF;                                                                 !
2560    4!        IF BDFR   <> 0 OR BMFR <> 0 THEN                                       !
2561    5!           [UFGDR] := [GSUBO(BC)] * [UFREQA];                                  !
2562    5!           CALL ROWPART ( [UFREQA], [UJK], , [PAJK] );                         !
2563    5!           CALL ROWMERGE ( [UFREQF], [UJK], [UFGDR], [PFJK] );                 !
2564    5!        ENDIF;                                                                 !
2565    4!     ELSE                                                                      !
2566    4!        IF NOMIT <> 0 THEN                                                     !
2567    5!$                                                                             $!
2568    5!$          DATA RECOVERY WITH STATIC CONDENSATION                             $!
2569    5!$                                                                             $!
2570    5!           PRINT("LOG=('          STATIC CONDENSATION RECOVERY')");           !
2571    5!           IF BLOAD <> 0 THEN                                                 !
2572    6!              CALL RECOVA ( [UA], [PO], [GSUBO(BC)], NRSET, [AA],             !
2573    6!                            [IFM(BC)], , [KOOINV(BC)],,[PFOA(BC)], [UF] );    !
2574    6!              IF NRSET <> 0 CALL RECOVA ( [AA], , [GSUBO(BC)],,,,,,,           !
2575    7!                                          [PFOA(BC)], [AF] );                  !
```

```
2576    6!            ENDIF;                                                    !
2577    5!            IF BSAERO <> 0 THEN                                       !
2578    6!               FOR S = 1 TO SUB DO                                    !
2579    7!                  CALL RECOVA ( [UAA(S)], [PAO(S)], [GASUBO(BC,S)],   !
2580    7!                                NRSET, [AAA(S)], [IFMA(BC,S)], BSAERO,!
2581    7!                                [KOOL(BC,S)], [KOOU(BC,S)],           !
2582    7!                                [PFOA(BC)], [UAFTMP] );               !
2583    7!$                                                                    $!
2584    7!$               MERGE THE CURRENT SUBCASE DEPENDENT RESULTS INTO A SINGLE  $!
2585    7!$               MATRIX OF RESPONSE QUANTITIES FOR FURTHER RECOVERY     $!
2586    7!$                                                                    $!
2587    7!                  CALL SAEROMRG ( BC, S, [UAF], [UAFTMP] );          !
2588    7!                  IF NRSET <> 0 THEN                                 !
2589    8!                     CALL RECOVA ( [AAA(S)],,[GASUBO(BC,S)],,,,,,,   !
2590    8!                                   [PFOA(BC)], [AAFTMP]);            !
2591    8!                     CALL SAEROMRG ( BC, S, [AAF], [AAFTMP] );       !
2592    8!                  ENDIF;                                             !
2593    7!               ENDDO;                                                !
2594    6!            ENDIF;                                                    !
2595    5!            IF BMODES <> 0 THEN                                       !
2596    6!               [PHIO] := [GSUBO(BC)] * [PHIA];                        !
2597    6!               CALL ROWMERGE ( [PHIF], [PHIO], [PHIA], [PFOA(BC)] );  !
2598    6!            ENDIF;                                                    !
2599    5!            IF BDTR <> 0 OR BMTR <> 0 THEN                            !
2600    6!               CALL RECOVA ( [UTRANA], , [GSUBO(BC)],,,,,,,           !
2601    6!                                         [PFOA(BC)], [UTRANF] );      !
2602    6!            ENDIF;                                                    !
2603    5!            IF BDFR <> 0 OR BMFR <> 0 THEN                            !
2604    6!               CALL RECOVA ( [UFREQA], , [GSUBO(BC)],,,,,,,           !
2605    6!                                         [PFOA(BC)], [UFREQF] );      !
2606    6!            ENDIF;                                                    !
2607    5!         ELSE                                                         !
2608    5!$                                                                   $!
2609    5!$            DATA RECOVERY WITHOUT F-SET REDUCTION                  $!
2610    5!$                                                                   $!
2611    5!            IF BLOAD <> 0 THEN                                        !
2612    6!               [UF] := [UA];                                          !
2613    6!               IF NRSET <> 0 [AF] := [AA];                            !
2614    6!            ENDIF;                                                    !
2615    5!            IF BSAERO <> 0 THEN                                       !
2616    6!               FOR S = 1 TO SUB DO                                    !
2617    7!$                                                                   $!
2618    7!$               MERGE THE CURRENT SUBCASE DEPENDENT RESULTS INTO A SINGLE  $!
2619    7!$               MATRIX OF RESPONSE QUANTITIES FOR FURTHER RECOVERY     $!
2620    7!$                                                                    $!
2621    7!                  CALL SAEROMRG ( BC, S, [UAF], [UAA(S)] );          !
2622    7!                  IF NRSET <> 0 CALL SAEROMRG ( BC, S, [AAF], [AAA(S)] );  !
2623    7!               ENDDO;                                                 !
2624    6!            ENDIF;                                                    !
2625    5!            IF BMODES <> 0 [PHIF]   := [PHIA];                        !
2626    5!            IF BDTR   <> 0 OR BMTR <> 0 [UTRANF] := [UTRANA];         !
2627    5!            IF BDFR   <> 0 OR BMFR <> 0 [UFREQF] := [UFREQA];         !
2628    5!         ENDIF;                                                       !
2629    4!      ENDIF;                                                          !
2630    3!$                                                                  $!
2631    3!      IF NBNDCOND > 1 CALL NULLMAT ( [UN], [AN], [PHIN] );            !
2632    3!      IF NSPC <> 0 THEN                                               !
2633    4!$                                                                  $!
2634    4!$         DATA RECOVERY WITH SPC-REDUCTION                         $!
2635    4!$                                                                  $!
2636    4!          PRINT("LOG=('          SPC RECOVERY')");                    !
2637    4!          IF BLOAD <> 0 THEN                                          !
2638    5!             CALL YSMERGE ( [UN], [YS(BC)], [UF], [PNSF(BC)] );       !
2639    5!             CALL OFPSPCF ( 0, BC, 1, 1, GSIZE, ESIZE(BC), NGDR,      !
2640    5!                            [KFS], [KSS], [UF], [YS(BC)], [PS],       !
2641    5!                            [PNSF(BC)], [PGMN(BC)], [PFJK], , , ,     !
2642    5!                            BGPDT(BC), OGRIDLOD );                    !
2643    5!             IF NRSET <> 0 CALL YSMERGE ( [AN], , [AF], [PNSF(BC)] );  !
2644    5!          ENDIF;                                                      !
2645    4!          IF BSAERO <> 0 THEN                                         !
2646    5!             CALL YSMERGE ( [UAN], [YS(BC)], [UAF], [PNSF(BC)] );    !
2647    5!             IF NRSET <> 0 CALL YSMERGE ( [AAN], , [AAF], [PNSF(BC)] );  !
2648    5!          ENDIF;                                                      !
2649    4!          IF BMODES <> 0 THEN                                         !
2650    5!             CALL YSMERGE ( [PHIN], [YS(BC)], [PHIF],                 !
2651    5!                                              [PNSF(BC)] );           !
2652    5!             IF DMODES <> 0   CALL OFPSPCF ( 0, BC, 2, 1, GSIZE,      !
2653    6!                                             ESIZE(BC), NGDR,         !
2654    6!                                             [KFS], , [PHIF], , ,     !
2655    6!                                             [PNSF(BC)], [PGMN(BC)], [PFJK],  !
2656    6!                                              , , , BGPDT(BC), OGRIDLOD );    !
```

```
2657    5!          ENDIF;                                                                !
2658    4!          IF BDTR   <> 0 OR BMTR <> 0                                           !
2659    5!                       CALL YSMERGE ( [UTRANN], [YS(BC)], [UTRANF],             !
2660    5!                                      [PNSF(BC)], BDTR );                       !
2661    4!          IF BDFR   <> 0 OR BMFR <> 0                                           !
2662    5!                       CALL YSMERGE ( [UFREQN], [YS(BC)], [UFREQF],             !
2663    5!                                      [PNSF(BC)], BDFR );                       !
2664    4!          IF BFLUTR <> 0                                                        !
2665    5!             CALL OFPSPCF ( 0, BC, 4, 2, GSIZE, ESIZE(BC), NGDR, [KFS], ,      !
2666    5!                            [PHIF], , , [PNSF(BC)], [PGMN(BC)], [PFJK],        !
2667    5!                             , , , BGPDT(BC), OGRIDLOD );                      !
2668    4!          IF BBLAST <> 0   THEN                                                 !
2669    5!             [UBLASTF] := [PHIF]*[UBLASTI];                                     !
2670    5!             CALL OFPSPCF ( 0, BC, 8, 1, GSIZE, ESIZE(BC), NGDR,                !
2671    5!                            [KFS], , [UBLASTF], , , [PNSF(BC)], [PGMN(BC)],     !
2672    5!                            [PFJK], , , , BGPDT(BC), OGRIDLOD );                !
2673    5!          ENDIF;                                                                !
2674    4!       ELSE                                                                     !
2675    4!$                                                                              $!
2676    4!$         DATA RECOVERY WITHOUT SPC-REDUCTION                                  $!
2677    4!$                                                                              $!
2678    4!          IF BLOAD <> 0 THEN                                                    !
2679    5!             [UN] := [UF];                                                      !
2680    5!             IF NRSET <> 0 [AN] := [AF];                                        !
2681    5!          ENDIF;                                                                !
2682    4!          IF BSAERO <> 0 THEN                                                   !
2683    5!             [UAN] := [UAF];                                                    !
2684    5!             IF NRSET <> 0 [AAN] := [AAF];                                      !
2685    5!          ENDIF;                                                                !
2686    4!          IF BMODES <> 0 [PHIN] := [PHIF];                                      !
2687    4!          IF BDTR   <> 0 OR BMTR <> 0 [UTRANN] := [UTRANA];                     !
2688    4!          IF BDFR   <> 0 OR BMFR <> 0 [UFREQN] := [UFREQA];                     !
2689    4!       ENDIF;                                                                   !
2690    3!$                                                                             $!
2691    3!       IF NBNDCOND > 1 CALL NULLMAT ( [UG(BC)], [AG(BC)], [UAG(BC)], [AAG(BC)],!
2692    4!                                      [PHIG(BC)] );                            !
2693    3!       IF NMPC <> 0 THEN                                                       !
2694    4!$                                                                             $!
2695    4!$         DATA RECOVERY WITH MPC-REDUCTION                                    $!
2696    4!$                                                                             $!
2697    4!          PRINT("LOG=('          MPC RECOVERY')");                            !
2698    4!          IF BLOAD <> 0 THEN                                                   !
2699    5!             [UM] := [TMN(BC)] * [UN];                                         !
2700    5!             CALL ROWMERGE ( [UG(BC)], [UM], [UN], [PGMN(BC)] );              !
2701    5!             IF NRSET <> 0 THEN                                                !
2702    6!                [UM] := [TMN(BC)] * [AN];                                      !
2703    6!                CALL ROWMERGE ( [AG(BC)], [UM], [AN], [PGMN(BC)] );           !
2704    6!             ENDIF;                                                            !
2705    5!          ENDIF;                                                               !
2706    4!          IF BSAERO <> 0 THEN                                                  !
2707    5!             [UM] := [TMN(BC)] * [UAN];                                        !
2708    5!             CALL ROWMERGE ( [UAG(BC)], [UM], [UAN], [PGMN(BC)] );            !
2709    5!             IF NRSET <> 0 THEN                                                !
2710    6!                [UM] := [TMN(BC)] * [AAN];                                     !
2711    6!                CALL ROWMERGE ( [AAG(BC)], [UM], [AAN], [PGMN(BC)] );         !
2712    6!             ENDIF;                                                            !
2713    5!          ENDIF;                                                               !
2714    4!          IF BMODES <> 0 THEN                                                  !
2715    5!             [UM] := [TMN(BC)] * [PHIN];                                       !
2716    5!             CALL ROWMERGE ( [PHIG(BC)], [UM], [PHIN], [PGMN(BC)] );          !
2717    5!          ENDIF;                                                               !
2718    4!          IF BDTR <> 0 OR BMTR <> 0 THEN                                       !
2719    5!             [UM] := [TMN(BC)] * [UTRANN];                                     !
2720    5!             CALL ROWMERGE ( [UTRANG], [UM], [UTRANN], [PGMN(BC)] );          !
2721    5!          ENDIF;                                                               !
2722    4!          IF BDFR <> 0 OR BMFR <> 0 THEN                                       !
2723    5!             [UM] := [TMN(BC)] * [UFREQN];                                     !
2724    5!             CALL ROWMERGE ( [UFREQG], [UM], [UFREQN], [PGMN(BC)] );          !
2725    5!          ENDIF;                                                               !
2726    4!       ELSE                                                                    !
2727    4!$                                                                             $!
2728    4!$         DATA RECOVERY WITHOUT MPC-REDUCTION                                 $!
2729    4!$                                                                             $!
2730    4!          IF BLOAD <> 0 THEN                                                   !
2731    5!             [UG(BC)] := [UN];                                                 !
2732    5!             IF NRSET <> 0 [AG(BC)] := [AN];                                   !
2733    5!          ENDIF;                                                               !
2734    4!          IF BSAERO <> 0 THEN                                                  !
2735    5!             [UAG(BC)] := [UAN];                                               !
2736    5!             IF NRSET <> 0 [AAG(BC)] := [AAN];                                 !
2737    5!          ENDIF;                                                               !
```

```
2738    4!          IF BMODES <> 0 [PHIG(BC)] := [PHIN];                                    !
2739    4!          IF BDTR   <> 0 OR BMTR <> 0 [UTRANG] := [UTRANN];                       !
2740    4!          IF BDFR   <> 0 OR BMFR <> 0 [UFREQG] := [UFREQN];                       !
2741    4!       ENDIF;                                                                     !
2742    3!$                                                                                $!
2743    3!$      RECOVER PHYSICAL BLAST DISCIPLINE DISPLACEMENTS                           $!
2744    3!$                                                                                $!
2745    3!       IF BBLAST <> 0 [UBLASTG] := [PHIG(BC)] * [UBLASTI];                       !
2746    3!$                                                                                $!
2747    3!$      HANDLE OUTPUT REQUESTS                                                    $!
2748    3!$                                                                                $!
2749    3!       PRINT("LOG=('          OUTPUT PROCESSING')");                             !
2750    3!       IF BSAERO <> 0 THEN                                                       !
2751    4!$                                                                               $!
2752    4!$         RECOVER STATIC AEROELASTIC LOADS DATA                                  $!
2753    4!$                                                                               $!
2754    4!          LOOP := TRUE;                                                          !
2755    4!          SUB  := 0;                                                             !
2756    4!          WHILE LOOP DO                                                          !
2757    5!             SUB := SUB + 1;                                                     !
2758    5!             CALL SAERODRV (BC, SUB, LOOP, MINDEX, SYM, MACH, QDP );             !
2759    5!$                                                                              $!
2760    5!$            CALL THE TRIMMED LOADS COMPUTATION WITH PROPER MATRICES           $!
2761    5!$                                                                              $!
2762    5!             IF SYM = 1  THEN                                                    !
2763    6!$***************************** TAKEN OUT FOR ZAERO ****************************$! <-
2764    6!$               CALL OFPALOAD ( , BC, MINDEX, SUB, GSIZE, BGPDT(BC),          $!
2765    6!$                               [GTKG], [GSTKG], QDP, [AIRFRC(MINDEX)],       $!
2766    6!$                               [DELTA(SUB)], [AICMAT(MINDEX)],               $!
2767    6!$                               [UAG(BC)], [MGG], [AAG(BC)], [KFS],           $!
2768    6!$                               [KSS], [UAF], [YS(BC)], [PNSF(BC)],           $!
2769    6!$                               [PGMN(BC)], [PFJK], NGDR, USET(BC),           $!
2770    6!$                               OGRIDLOD );                                   $!
2771    6!$*************************************************************************$!
2772    6!               CALL OFPALOAD ( , BC, MINDEX, SUB, GSIZE, BGPDT(BC),           !
2773    6!                               [UGTKG], [UGTKG], QDP, [AIRFRC(MINDEX)],       !
2774    6!                               [DELTA(SUB)], [AICMAT(MINDEX)],                !
2775    6!                               [UAG(BC)], [MGG], [AAG(BC)], [KFS],            !
2776    6!                               [KSS], [UAF], [YS(BC)], [PNSF(BC)],            !
2777    6!                               [PGMN(BC)], [PFJK], NGDR, USET(BC),            !
2778    6!                               OGRIDLOD );                                    ! <-
2779    6!             ELSE                                                              !
2780    6!                IF SYM = -1 THEN                                               !
2781    7!$***************************** TAKEN OUT FOR ZAERO ****************************$! <-
2782    7!$                  CALL OFPALOAD ( , BC, MINDEX, SUB, GSIZE, BGPDT(BC),       $!
2783    7!$                                  [GTKG], [GSTKG], QDP, [AIRFRC(MINDEX)],    $!
2784    7!$                                  [DELTA(SUB)], [AAICMAT(MINDEX)],           $!
2785    7!$                                  [UAG(BC)], [MGG], [AAG(BC)], [KFS],        $!
2786    7!$                                  [KSS], [UAF], [YS(BC)], [PNSF(BC)],        $!
2787    7!$                                  [PGMN(BC)], [PFJK], NGDR, USET(BC),        $!
2788    7!$                                  OGRIDLOD );                                $!
2789    7!$*************************************************************************$!
2790    7!                   CALL OFPALOAD ( , BC, MINDEX, SUB, GSIZE, BGPDT(BC),        !
2791    7!                                   [UGTKG], [UGTKG], QDP, [AIRFRC(MINDEX)],    !
2792    7!                                   [DELTA(SUB)], [AAICMAT(MINDEX)],            !
2793    7!                                   [UAG(BC)], [MGG], [AAG(BC)], [KFS],         !
2794    7!                                   [KSS], [UAF], [YS(BC)], [PNSF(BC)],         !
2795    7!                                   [PGMN(BC)], [PFJK], NGDR, USET(BC),         !
2796    7!                                   OGRIDLOD );                                 ! <-
2797    7!                ENDIF;                                                         !
2798    6!             ENDIF;                                                            !
2799    5!$                                                                             $!
2800    5!$            CALL TO COMPUTE THE TRIMMED LOADS/DISPLACEMENTS ON THE           $!
2801    5!$            AERODYNAMIC MODEL                                                $!
2802    5!$                                                                             $!
2803    5!             IF SYM = 1  THEN                                                  !
2804    6!$***************************** TAKEN OUT FOR ZAERO ****************************$! <-
2805    6!$               CALL OFPAEROM ( NITER, BC, MINDEX, SUB, GSIZE, GEOMSA,       $!
2806    6!$                               [GTKG], [GSTKG], QDP, [AIRFRC(MINDEX)],      $!
2807    6!$                               [DELTA(SUB)], [AICMAT(MINDEX)],              $!
2808    6!$                               [UAG(BC)], OAGRDLOD, OAGRDDSP );             $!
2809    6!$*************************************************************************$!
2810    6!               CALL OFPAEROM ( NITER, BC, MINDEX, SUB, GSIZE, GEOMSA,         !
2811    6!                               [UGTKG], [UGTKG], QDP, [AIRFRC(MINDEX)],       !
2812    6!                               [DELTA(SUB)], [AICMAT(MINDEX)],                !
2813    6!                               [UAG(BC)], OAGRDLOD, OAGRDDSP );               ! <-
2814    6!             ELSE                                                              !
2815    6!                IF SYM = -1 THEN                                               !
2816    7!$***************************** TAKEN OUT FOR ZAERO ****************************$! <-
2817    7!$                  CALL OFPAEROM ( NITER, BC, MINDEX, SUB, GSIZE, GEOMSA,     $!
2818    7!$                                  [GTKG], [GSTKG], QDP, [AIRFRC(MINDEX)],    $!
```

```
2819    7!$                                [DELTA(SUB)], [AAICMAT(MINDEX)],                $!
2820    7!$                                [UAG(BC)], OAGRDLOD, OAGRDDSP );                $!
2821    7!$*************************************************************************************$!
2822    7!                  CALL OFPAEROM ( NITER, BC, MINDEX, SUB, GSIZE, GEOMSA,           !
2823    7!                                  [UGTKG], [UGTKG], QDP, [AIRFRC(MINDEX)],         !
2824    7!                                  [DELTA(SUB)], [AAICMAT(MINDEX)],                 !
2825    7!                                  [UAG(BC)], OAGRDLOD, OAGRDDSP );                 !
2826    7!              ENDIF;                                                               !
2827    6!            ENDIF;                                                                 !
2828    5!          ENDDO;                                                                   !
2829    4!        ENDIF;                                                                     !
2830    3!        IF BDRSP <> 0  THEN                                                        !
2831    4!          CALL OFPDLOAD ( , BC, BGPDT(BC), PSIZE(BC), ESIZE(BC), [PHIG(BC)],       !
2832    4!                          [PTGLOAD], [PTHLOAD], [PFGLOAD], [PFHLOAD], OGRIDLOD );  !
2833    4!          IF BDTR   <> 0 OR BMTR <> 0                                              !
2834    5!                         CALL OFPSPCF ( 0, BC, 5, 1, GSIZE, ESIZE(BC),             !
2835    5!                                        NGDR, [KFS], , [UTRANF], , ,               !
2836    5!                                        [PNSF(BC)], [PGMN(BC)], [PFJK],            !
2837    5!                                        [PHIG(BC)], [PTGLOAD], [PTHLOAD],          !
2838    5!                                        BGPDT(BC), OGRIDLOD );                     !
2839    4!          IF BDFR   <> 0 OR BMFR <> 0                                              !
2840    5!                         CALL OFPSPCF ( 0, BC, 6, 2, GSIZE, ESIZE(BC),             !
2841    5!                                        NGDR, [KFS], , [UFREQF], , ,               !
2842    5!                                        [PNSF(BC)], [PGMN(BC)], [PFJK],            !
2843    5!                                        [PHIG(BC)], [PFGLOAD], [PFHLOAD],          !
2844    5!                                        BGPDT(BC), OGRIDLOD );                     !
2845    4!        ENDIF;                                                                     !
2846    3!        CALL OFPLOAD ( NUMOPTBC, BC, , GSIZE, BGPDT(BC), PSIZE(BC),              !
2847    3!                       [PG] );                                                     !
2848    3!        CALL OFPDISP( NUMOPTBC, BC, , GSIZE, BGPDT(BC), ESIZE(BC), PSIZE(BC),    !
2849    3!                      OGRIDDSP, [UG(BC)], [AG(BC)], [UAG(BC)], [AAG(BC)],        !
2850    3!                      [UBLASTG], , [UTRANG], [UTRANE], [UFREQG], [UFREQE],       !
2851    3!                      LAMBDA, [PHIG(BC)] );                                      !
2852    3!        CALL EDR ( NUMOPTBC, BC, , NDV, GSIZE, EOSUMMRY, EODISC,                 !
2853    3!                   GLBDES, LOCLVAR, [PTRANS],                                    !
2854    3!                   [UG(BC)], [UAG(BC)], , [UTRANG], [UFREQG], [PHIG(BC)] );      !
2855    3!        CALL OFPEDR ( BC, HSIZE(BC) );                                           !
2856    3!     ENDDO;                                                                      !
2857    2!ENDIF;                                                                         !
2858    1!END;                                                                           !
```

APPENDIX C

ZAERO BULK DATA TEMPLATE DEFINITIONS

(TEMPLATE.DAT)

The following lists the twenty three (23) new bulk data templates in file (**TEMPLATE.DAT**) used to define the ZAERO bulk data cards:

```
ACOORD |ID      |XORIGN |YORIGN |ZORIGN |DELTA  |THETA  |XMCNT  |YMCNT  |CONT   |
CHAR    INT      REAL    REAL    REAL    REAL    REAL    REAL    REAL    CHAR
DEFAULT          0.0     0.0     0.0     0.0     0.0     0.0     0.0
CHECKS  GT 0
        1        2       3       4       5       6       7       8
ACOORD  ID       XORIGN  YORIGN  ZORIGN  DELTA   THETA   XMCNT   YMCNT
+COORD |ZMCNT  |XBEND  |YBEND  |ZBEND  |XTORQ  |YTORQ  |ZTORQ  |       |       |
CHAR    REAL     REAL    REAL    REAL    REAL    REAL    REAL
DEFAULT 0.0      0.0     0.0     0.0     0.0     0.0     0.0
CHECKS
        9        10      11      12      13      14      -15
        ZMCNT    XBEND   YBEND   ZBEND   XTORQ   YTORQ   ZTORQ                   $
```

```
AEROZ  |ACSID |XZSYM  |RHOREF |REFC   |REFB   |REFS   |GREF   |       |       |
CHAR    INT     CHAR    REAL    REAL    REAL    REAL    INT
DEFAULT 0       YES     1.0     1.0     1.0     1.0     0
CHECKS                  GE 0.   GE 0.   GE 0.   GE 0.   GE 0
        1       2       3       4       5       6       -7
AEROZ   ACSID   XZSYM   RHOREF  REFC    REFB    REFS    GREF                    $
```

```
AESURFZ|LABEL  |TYPE   |CID    |SETK   |SETG   |       |       |       |       |
CHAR    CHAR    CHAR    INT     INT     INT
DEFAULT
CHECKS                  GE 0    GT 0    GE 0
        1       3       5       6       -7
AESURFZ LABEL   TYPE    CID     SETK    SETG                                    $
```

```
ATTACH |EID    |MODEL  |SETK   |REFGRID|FEEDBK |       |       |       |       |
CHAR    INT     CHAR    INT     INT     CHAR
DEFAULT                                 FLEX
CHECKS  GT 0            GT 0    GT 0    FRCHK
        1       2       4       5       -6
ATTACH  EID     MODEL   BOXSETIDREFGRD  FEEDBK                                  $
```

```
BODY7  |IDBODY |LABELB |IPBODY |ACOORD |NSEG   |ID(1)  |ID(2)  |ID(3)  |CONT   |
CHAR    INT     CHAR    INT     INT     INT     INT     INT     INT     CHAR
DEFAULT                 0       0                       NULL    NULL
CHECKS  GT 0            GE 0    GE 0    GE 1    GT 0    GTZOB   GTZOB
        1       2       4       5       6       7       8       9
BODY7   IDBODY  LABELB  IPBODY  ACOORD  NSEG    IDMESHA IDMESHB IDMESHC
+BODY7 |ID(5)  |ID(6)  |ID(7)  |ID(8)  |ID(9)  |ID(10) |ID(11) |ID(12) |       |
CHAR    INT     INT     INT     INT     INT     INT     INT     INT
DEFAULT NULL    NULL    NULL    NULL    NULL    NULL    NULL    NULL
CHECKS  GTZOB   GTZOB   GTZOB   GTZOB   GTZOB   GTZOB   GTZOB   GTZOB
        10      11      12      13      14      15      16      -17
        IDMESHD IDMESHE IDMESHF IDMESHG IDMESHH IDMESHI IDMESHJ IDMESHK         $
```

```
CAERO7 |EID     |LABELC |ACOORD |NSPAN  |NCHORD |LSPAN  |ZTAIC  |PAFOIL |CONT   |
CHAR    INT      CHAR    INT     INT     INT     INT     INT     INT     CHAR
DEFAULT                  0                       0       0       0
CHECKS  GT 0             GE 0    GE 2    GE 2    GE 0    GE 0    GE 0
        1        2       4       5       6       7       8       9
CAERO7  EID      LABELC  ACOORD  NSPAN   NCHORD  LSPAN   ZTAIC   PAFOIL
+CAERO7|XRL     |YRL    |ZRL    |RCH    |LRCHD  |ATTR   |       |       |CONT   |
CHAR    REAL     REAL    REAL    REAL    INT     INT                     CHAR
DEFAULT                                  0       0
CHECKS                           GE 0.   GE 0    GE 0
        10       11      12      13      14      15
        XRL      YRL     ZRL     RCH     LRCHD   ATTR
+CAERO7|XTL     |YTL    |ZTL    |TCH    |LTCHD  |ATTT   |       |       |       |
CHAR    REAL     REAL    REAL    REAL    INT     INT
DEFAULT                                  0       0
CHECKS                           GE 0.   GE 0    GE 0
        16       17      18      19      20      -21
        XTL      YTL     ZTL     TCH     LTCHD   ATTT                            $
```

```
CHORDCP|ID      |X      |CPU    |CPL    |X      |CPU    |CPL    |       |CONT   |
CHAR    INT      REAL    REAL    REAL    REAL    REAL    REAL            CHAR
DEFAULT
CHECKS  GT 0     GE 0.                   GE 0.
        1        2       3       -4      2       3       -4
CHORDCP ID       X       CPU     CPL
+CHRDCP|        |X      |CPU    |CPL    |X      |CPU    |CPL    |       |ETC    |
CHAR            REAL    REAL    REAL    REAL    REAL    REAL            CHAR
DEFAULT
CHECKS          GE 0.                   GE 0.
                2       3       -4      2       3       -4
                                                                        $
```

```
FLUTTER|SID     |METHOD |DENS   |IDMK   |VEL    |MLIST  |KLIST  |EFFID  |CONT   |
CHAR    INT      CHAR    INT     INT     INT     INT     INT     INT
DEFAULT          PK
CHECKS  GT 0             GT 0    GT 0    GT 0    GE 0    GE 0    GE 0
        1        2       3       4       5       6       7       8
FLUTTER SETID    METHOD  DENS    IDMK    VEL     MLIST   KLIST   EFFID
+FLUTTR|SYMXZ   |SYMXY  |EPS    |CURVFIT|PRINT  |
CHAR    INT      INT     REAL    CHAR    INT
DEFAULT                  1.E-5   LINEAR  0
CHECKS  IB -1            GT 0.   FLTFIT
        9  1     10      11      12      -14
        SYMXZ    SYMXY   EPS     CURVFIT PRINT                           $
```

```
GUST   |SID     |GLOAD  |WG     |X0     |V      |QDP    |IDMK   |       |CONT   |
CHAR    INT      INT     REAL    REAL    REAL    REAL    INT             CHAR
DEFAULT
CHECKS  GT 0     GT 0    NE 0.           GT 0.   GT 0.   GT 0
        1        2       3       4       5       6       7
GUST    SID      GLOAD   WG      X0      V       QDP     IDMK
+GUST  |SYMXZ   |SYMXY  |
CHAR    INT      INT
DEFAULT
CHECKS  IB -1
        8  1     -9 0
        SYMXZ    SYMXY                                                   $
```

```
LOADMOD|LID     |LABEL  |CP     |SETK   |SETG   |       |       |       |       |
CHAR    INT      CHAR    INT     INT     INT
DEFAULT
CHECKS  GT 0             GE 0    GT 0    GT 0
        1        2       4       5       -6
LOADMOD LID      LABEL   CP      SETK    SETG                            $
```

```
MACHCP |ID     |MACH   |IGRID  |INDICIA|SPNID  |CHDCP  |SPNID  |CHDCP  |CONT   |
CHAR    INT     REAL    INT     INT     INT     INT     INT     INT     CHAR
DEFAULT         0.9     0       0
CHECKS  GT 0    GE 0.           GE 0    GT 0    GT 0
        1       2       3       4       5       -6      5       -6
MACHCP  ID      MACH    IGRID   INDICIA SPANID  CHORDCP
+MACHCP|SPNID   |CHDCP  |SPNID  |CHDCP  |SPNID  |CHDCP  |SPNID  |CHDCP  |ETC    |
CHAR    INT     INT     INT     INT     INT     INT     INT     INT     CHAR
DEFAULT
CHECKS
        5       -6      5       -6      5       -6      5       -6
                                                                        $
```

```
MKAEROZ|IDMK   |MACH   |METHOD |IDFLT  |SAVE   |FILE1  |FILE2  |PRINT  |CONT   |
CHAR    INT     REAL    INT     INT     CHAR    CHAR    CHAR    INT     CHAR
DEFAULT                 0       0
CHECKS  GT 0    GE 0.0          GE 0
        1       2       3       4       5       7       9       11
MKAEROZ IDMK    MACH    METHOD  IDFLT   SAVE    FILE1   FILE2   PRINT
+MKAROZ|FREQ(1)|FREQ(2)|FREQ(3)|FREQ(4)|FREQ(5)|FREQ(6)|FREQ(7)|FREQ(9)|ETC    |
CHAR    REAL    REAL    REAL    REAL    REAL    REAL    REAL    REAL    CHAR
DEFAULT 0.
CHECKS  GE 0.   GE 0.   GE 0.   GE 0.   GE 0.   GE 0.   GE 0.   GE 0.
        -12     -12     -12     -12     -12     -12     -12     -12
        RFREQ                                                           $
```

```
PAFOIL7|ID     |IAFX   |ITHR   |ICAMR  |RADR   |ITHT   |ICAMT  |RADT   |       |
CHAR    INT     INT     INT     INT     REAL    INT     INT     REAL
DEFAULT         0       0       0       0.0     0       0       0.0
CHECKS  GT 0            GE 0    GE 0    GE 0.0  GE 0    GE 0    GE 0.0
        1       2       3       4       5       6       7       -8
PAFOIL7 ID      IAFX    ITHR    ICAMR   RADR    ITHT    ICAMT   RADT          $
```

```
PANLST1|SETID  |MACROID|BOX1   |BOX2   |       |       |       |       |       |
CHAR    INT     INT     INT     INT
DEFAULT
CHECKS  GT 0    GT 0    GT 0    GEP
        1       2       3       -4
PANLST1 SETID   MACROID BOX1    BOX2                                          $
```

```
PANLST2|SETID  |MACROID|B(1)   |B(2)   |B(3)   |B(4)   |B(5)   |B(6)   |CONT   |
CHAR    INT     INT     INT     INT/CHARINT     INT     INT     INT     CHAR
DEFAULT
CHECKS  GT 0    GT 0    GE 0    GE 0    GE 0    GE 0    GE 0    GE 0
        1       2       -3      -3      -3      -3      -3      -3
PANLST2 SETID   MACROID BOXI
+PNLST2|B(N)   |B(N+1) |B(N+2) |B(N+3) |B(N+4) |B(N+5) |B(N+6) |B(N+7) |ETC    |
CHAR    INT     INT     INT     INT     INT     INT     INT     INT     CHAR
DEFAULT
CHECKS  GE 0    GE 0    GE 0    GE 0    GE 0    GE 0    GE 0    GE 0
        -3      -3      -3      -3      -3      -3      -3      -3
                                                                        $
```

```
PBODY7 |IPBODY7|WAKE    |CPBASE |XSWAKE |XDWAKE |YWAKE  |ZWAKE  |INLET  |CONT   |
CHAR    INT     INT     REAL    REAL    REAL    REAL    REAL    INT     CHAR
DEFAULT         0       -0.2    1.3     1.1     0.0     0.0     0
CHECKS  GT 0    GE 0            GE 1.0  GE 1.0                  GE 0
        1       2       3       4       5       6       7       8
PBODY7  IPBODY7 WAKE    CPBASE  XSWAKE  XDWAKE  YWAKE   ZWAKE   INLET
+PBODY7|IDP(1) |FLW(1) |IDP(2) |FLW(2) |IDP(3) |FLW(3) |IDP(4) |FLW(4) |ETC    |
CHAR    INT     REAL    INT     REAL    INT     REAL    INT     REAL    CHAR
DEFAULT 0       0.0     0       0.0     0       0.0     0       0.0
CHECKS
        9       -10     9       -10     9       -10     9       -10
        IDP     FLOWRT                                                  $
```

```
SEGMESH|IDMESH |NAXIS  |NRAD   |       |       |       |       |       |CONT   |
CHAR    INT     INT     INT                                             CHAR
DEFAULT
CHECKS  GT 0    GE 2    GE 2
        1       2       3
SEGMESH IDMESH  NAXIS   NRAD
+SEG1  |IT(N)  |X(N)   |CM(N)  |YR(N)  |ZR(N)  |IY(N)  |IZ(N)  |       |ETCT   |
CHAR    INT     REAL    REAL    REAL    REAL    INT     INT             CHAR
DEFAULT                                         0       0
CHECKS  GT 0
        4       5       6       7       8       9       -10
        ITYPE   X       CAM     YR      ZR      IDY     IDZ                     $
```

```
SPLINE1|EID    |MODEL  |CP     |SETK   |SETG   |DZ     |EPS    |       |       |
CHAR    INT     CHAR    INT     INT     INT     REAL    REAL
DEFAULT                                                 0.01
CHECKS  GT 0            GE 0    GT 0    GT 0    GE 0.   GE 0.
        1       2       4       5       6       7       -8
SPLINE1 EID     MODEL   CP      BOXSETIDGRDSETIDFLEX    EPS                     $
```

```
SPLINE2|EID    |MODEL  |SETK   |SETG   |DZ     |DTOR   |CID    |DTHX   |CONT   |
CHAR    INT     CHAR    INT     INT     REAL    REAL    INT     REAL    CHAR
DEFAULT
CHECKS  GT 0            GT 0    GT 0    GE 0.   GE 0.   GE 0
        1       2       4       5       6       7       8       9
SPLINE2 EID     MODEL   BOXSETIDGRDSETIDFLEX    DTOR    CID     DTHX
+SPLNE2|DTHY   |       |       |       |       |       |       |       |
CHAR    REAL
DEFAULT    .
CHECKS
        -10
        DTHY                                                            $
```

```
SPLINE3|EID    |MODEL  |CP     |SETK   |SETG   |DZ     |EPS    |       |       |
CHAR    INT     CHAR    INT     INT     INT     REAL    REAL
DEFAULT                                                 0.01
CHECKS  GT 0            GE 0    GT 0    GT 0    GE 0.   GE 0.
        1       2       4       5       6       7       -8
SPLINE3 EID     MODEL   CP      BOXSETIDGRDSETIDFLEX    EPS                     $
SPOINT |ID     |ID     |ID     |ID     |ID     |ID     |ID     |ID     |       |
CHAR    INT     INT/CHARINT     INT     INT     INT     INT     INT
DEFAULT
CHECKS  GT 0    GE 0    GE 0    GE 0    GE 0    GE 0    GE 0    GE 0
        -1      -1      -1      -1      -1      -1      -1      -1
SPOINT  EXTID                                                           $
```

```
TRIM    |SETID  |IDMK   |QDP    |TRMTYP |EFFID  |V0     |PRINT  |       |CONT   |
CHAR     INT     INT     REAL    CHAR    INT     REAL    INT             CHAR
DEFAULT
CHECKS  GT 0    GT 0    GT 0.   TRIM    GE 0    GE 0.
        1       2       3       4       6       7       8
TRIM    SETID   IDMK    QDP     TRMTYP  EFFID   V0      PRINT
+TRIM   |LABEL1 |VALUE1 |LABEL2 |VALUE2 |LABEL3 |VALUE3 |LABEL4 |VALUE4 |ETC    |
CHAR     CHAR    REL/CHARCHAR    REL/CHARCHAR    REL/CHARCHAR    REL/CHARCHAR
DEFAULT
CHECKS
        9       -11     9       -11     9       -11     9       -11
        LABELI  FIXI    FREEI                                                   $
```

```
TRIMFLT|IDFLT  |TILTA  |ALPHA  |BETA   |PRATE  |QRATE  |RRATE  |       |CONT   |
CHAR     INT     INT     REAL    REAL    REAL    REAL    REAL            CHAR
DEFAULT         0       0.0     0.0     0.0     0.0     0.0
CHECKS  GT 0
        1       2       3       4       5       6       7
TRIMFLT IDFLT   TILTA   ALPHA   BETA    PRATE   QRATE   RRATE
+TRIMF  |LABEL1 |VALUE1 |LABEL2 |VALUE2 |LABEL3 |VALUE3 |LABEL4 |VALUE4 |ETC    |
CHAR     CHAR    REAL    CHAR    REAL    CHAR    REAL    CHAR    REAL    CHAR
DEFAULT         0.0
CHECKS
        8       -10     8       -10     8       -10     8       -10
        LABELI  VALUE                                                           $
```

```
ZTAIC   |ID     |NFLAP  |MACHCP1|MACHCP2|MACHCP3|MACHCP4|MACHCP5|MACHCP6|CONT   |
CHAR     INT     INT     INT     INT     INT     INT     INT     INT     CHAR
DEFAULT         0               0       0       0       0       0
CHECKS  GT 0    GE 0    GT 0    GE 0    GE 0    GE 0    GE 0    GE 0
        1       2       3       4       5       6       7       8
ZTAIC   ID      NFLAP   MACHCP1 MACHCP2 MACHCP3 MACHCP4 MACHCP5 MACHCP6
+ZTAIC  |LABEL  |HINGE  |INBDY  |OUTBDY |LABEL  |HINGE  |INBDY  |OUTBDY |ETC    |
CHAR     CHAR    INT     INT     INT     CHAR    INT     INT     INT     CHAR
DEFAULT         2       1       2
CHECKS          GE 1    GE 1    GE 2
        9       10      11      -12     9       10      11      -12
        LABEL   HINGE   INBDY   OUTBDY                                          $
```

APPENDIX D

ZAERO RELATIONAL SCHEMA DEFINITION

(RELATION.DAT)

The following are the relational SCHEMA definitions (from file RELATION.DAT) for all database relational entities used by the ZAERO module:

```
RELATION ACOORD
ID        INT
XORIGN    RSP
YORIGN    RSP
ZORIGN    RSP
DELTA     RSP
THETA     RSP
XMCNT     RSP
YMCNT     RSP
ZMCNT     RSP
XBEND     RSP
YBEND     RSP
ZBEND     RSP
XTORQ     RSP
YTORQ     RSP
ZTORQ     RSP
END
-------------------------
RELATION AGRIDZ
EXTID     INT
INTID     INT
CORD      INT
X         RSP
Y         RSP
Z         RSP
END
-------------------------
RELATION AEROZ
ACSID     INT
XZSYM     STR     4
RHOREF    RSP
REFC      RSP
REFB      RSP
REFS      RSP
GREF      INT
END
-------------------------
RELATION AESURFZ
LABEL     STR     8
TYPE      STR     8
CID       INT
SETK      INT
SETG      INT
END
-------------------------
RELATION AQUADZ
MACROID   INT
ACMPNT    INT
NDOF      INT
EXTID     INT
INTID     INT
AREA      RSP
X         RSP
Y         RSP
Z         RSP
N1        RSP
N2        RSP
N3        RSP
R1        RSP
R2        RSP
R3        RSP
RTHETA    RSP
RDELTA    RSP
CHORD     RSP
ID1       RSP
ID2       RSP
ID3       RSP
ID4       RSP
CAM85     RSP
CAM95     RSP
DZX85     RSP
DZX95     RSP
DZXLE     RSP
DZXTE     RSP
END
```

```
RELATION BODY7
IDBODY    INT
LABELB    STR     8
IPBODY    INT
ACOORD    INT
NSEG      INT
IDMESHA   INT
IDMESHB   INT
IDMESHC   INT
IDMESHD   INT
IDMESHE   INT
IDMESHF   INT
IDMESHG   INT
IDMESHH   INT
IDMESHI   INT
IDMESHJ   INT
IDMESHK   INT
END
-------------------------
RELATION CAERO7
EID       INT
LABELC    STR     8
ACOORD    INT
NSPAN     INT
NCHORD    INT
LSPAN     INT
ZTAIC     INT
PAFOIL    INT
XRL       RSP
YRL       RSP
ZRL       RSP
RCH       RSP
LRCHD     INT
ATTR      INT
XTL       RSP
YTL       RSP
ZTL       RSP
TCH       RSP
LTCHD     INT
ATTT      INT
END
-------------------------
RELATION CHORDCP
ID        INT
X         RSP
CPU       RSP
CPL       RSP
END
-------------------------
RELATION FLUTTER
SETID     INT
METHOD    STR     4
DENS      INT
IDMK      INT
VEL       INT
MLIST     INT
KLIST     INT
EFFID     INT
SYMXZ     INT
SYMXY     INT
EPS       RSP
CURVFIT   STR     8
MACHVAL   RSP
PRINT     INT
END
-------------------------
```

```
RELATION GEOMZA

MACROID   INT
ACMPNT    STR     8
NDOF      INT
EXTID     INT
INTID     INT
AREA      RSP
X         RSP
Y         RSP
Z         RSP
N1        RSP
N2        RSP
N3        RSP
R1        RSP
R2        RSP
R3        RSP
RTHETA    RSP
RDELTA    RSP
CHORD     RSP
ID1       RSP
ID2       RSP
ID3       RSP
ID4       RSP
CAM85     RSP
CAM95     RSP
DZX85     RSP
DZX95     RSP
DZXLE     RSP
DZXTE     RSP
INLET     INT
IWAKE     INT
END
-------------------------
RELATION LOADMOD
LID       INT
LABEL     STR     8
CP        INT
SETK      INT
SETG      INT
END
-------------------------
RELATION MACHCP
ID        INT
MACH      RSP
IGRID     INT
INDICIA   INT
SPANID    INT
CHORDCP   INT
END
-------------------------
RELATION MKAEROZ
IDMK      INT
MACH      RSP
METHOD    INT
IDFLT     INT
SAVE      STR     8
FILE1     STR     8
FILE2     STR     8
PRINT     INT
RFREQ     RSP
END
-------------------------
RELATION PAFOIL7
ID        INT
IAFX      INT
ITHR      INT
ICAMR     INT
RADR      RSP
ITHT      INT
ICAMT     INT
RADT      RSP
END
```

```
RELATION PANLST1
SETID     INT
MACROID   INT
BOX1      INT
BOX2      INT
END
-------------------------
RELATION PANLST2
SETID     INT
MACROID   INT
BOXI      INT
END
-------------------------
RELATION PBODY7
IPBODY7   INT
WAKE      INT
CPBASE    RSP
XSWAKE    RSP
XDWAKE    RSP
YWAKE     RSP
ZWAKE     RSP
INLET     INT
IDP       INT
FLOWRT    RSP
END
-------------------------
RELATION REUNMK
IDMK      INT
MACH      RSP
METHOD    INT
SYMXZ     INT
ALPHA     RSP
BETA      RSP
PRATE     RSP
QRATE     RSP
RRATE     RSP
MINDEX    INT
KINDEX    INT
RFREQ     RSP
END
-------------------------
RELATION SEGMESH
IDMESH    INT
NAXIS     INT
NRAD      INT
ITYPE     INT
X         RSP
CAM       RSP
YR        RSP
ZR        RSP
IDY       INT
IDZ       INT
END
-------------------------
RELATION SPLINE3
EID       KINT
MODEL     STR     8
CP        INT
BOXSETID  INT
GRDSETID  INT
FLEX      RSP
EPS       RSP
END
-------------------------
RELATION TRIMFLT
IDFLT     INT
TILTA     INT
ALPHA     RSP
BETA      RSP
PRATE     RSP
QRATE     RSP
RRATE     RSP
LABELI    STR     8
VALUE     RSP
END
```

```
RELATION ZTAIC
ID       INT
NFLAP    INT
MACHCP1  INT
MACHCP2  INT
MACHCP3  INT
MACHCP4  INT
MACHCP5  INT
MACHCP6  INT
LABEL    STR    4
HINGE    INT
INBDY    INT
OUTBDY   INT
END
```

APPENDIX E

ZAERO ERROR MESSAGE DEFINITION

(SERRMSG.DAT)

In following the ASTROS format for error message definitions, three new error message modules (numbers 35 through 37) have been generated for the ZAERO software and added to the SERRMSG.DAT file. These ZAERO error message modules are listed as follows:

```
*MODULE 35     ZONA'S AEROGM MODULE MESSAGES
'NO $ BULK DATA ENTRIES ARE DEFINED, BUT BODY7 BULK DATA EXISTS IN THE INPUT.'
'$ BULK DATA ENTRY WITH BID: $ HAS $ NUMBER OF SEGMENTS, BUT THERE ARE ONLY $ NUMBER OF SEGMESH '
'BULK DATA ENTRIES DEFINED.'
'BULK DATA ENTRY $ IS REFERED BY A ID: $ BUT NO $ EXISTS IN THE INPUT.'
'ID NUMBER: $ OF BULK DATA CARD $ IS NOT DEFINED.'
'BULK DATA ENTRY $ WITH ID: $ , REFERS TO BULK DATA ENTRY $ WITH ID: $ WHICH DOES NOT EXIST.'
'$ BULK DATA CARD WITH ID: $ SPECIFIES $ NUMBER OF AXIAL STATIONS, BUT ONLY $ ARE DEFINED.'
'THE X-LOCATIONS OF A $ BULK DATA ENTRY WITH IDMESH: $ ARE NOT IN ASCENDING ORDER AT AXIAL STATIONS $ AND $.'
'$ WITH WID: $ HAS $ NUMBER OF SPANWISE DIVISIONS DEFINED, BUT THERE ARE $ NUMBER OF VALUES'
'LISTED IN THE CORRESPONDING $ BULK DATA ENTRY WITH ID: $ .'
'$ WITH ID: $ REFERENCED BY $ WITH WID: $ DOES NOT BEGIN WITH 0.0 OR END AT 100.0.'
'THE SPANWISE DIVISIONS OF A $ BULK DATA CARD, ID: $ REFERENCED BY A $ CARD WITH WID: $, ARE NOT'
'IN ASCENDING ORDER.'
'THE TOTAL NUMBER OF MACH NUMBERS LISTED IN ALL MACHCP BULK DATA ENTRIES EXCEEDS 6.'
'CAERO7 ENTRY WITH WID: $, HAS NO STEADY PRESSURE INPUT ON SPANWISE STRIP INDEX = $ AND MACH NUMBER = $.'
'THEREFORE LINEAR UNSTEADY PRESSURE WILL BE COMPUTED FOR THIS STRIP.'
'CAERO7 ENTRY WITH WID: $, HAS MORE THAN ONE SPANWISE STRIP INDEX DEFINED FOR A MACHCP BULK DATA ENTRY'
'FOR SPANWISE STRIP INDEX = $ AND MACH NUMBER = $.'
'AERODYNAMIC $ ID: $ IS TOO LARGE BASED ON AVAILABLE MEMORY.'
'A DUPLICATE AERODYNAMIC $ EXISTS WITH ID: $ .'
'A SEGMESH BULK DATA CARD WITH IDMESH: $ HAS $ NUMBER OF $-VALUE CIRCUMFERENTIAL POINTS (NRAD) DEFINED,'
'BUT THERE ARE ONLY $ NUMBER OF VALUES LISTED IN AEFACT WITH ID: $.'
'A $ WITH ID: $ HAS A BOX OF ZERO AREA WITH ID: $.'
'ERROR IN $ WITH ID: $. INCOMPLETE LIST OF LABEL-HINGE-INBDY-OUTBDY PAIRS FOR NFLAP = $.'
'ERROR IN $ WITH ID: $. ENTRY LABEL = $ IS NOT $ OR $.'
'ERROR IN $ WITH ID: $. ENTRY HINGE = $ IS NOT GREATER THAN 1 AND LESS THAN $ (NCHORD).'
'ERROR IN $ WITH ID: $. ENTRY INBDY = $ IS NOT GREATER OR EQUAL TO 1 AND LESS THAN $ (NSPAN).'
'ERROR IN $ WITH ID: $. ENTRY OUTBDY = $ IS NOT GREATER THAN 1 AND LESS THAN OR EQUAL TO $ (NSPAN).'
'ERROR IN $ WITH ID: $. ENTRY INBDY = $ IS GREATER THAN OR EQUAL TO ENTRY OUTBDY = $.'
'A $ BULK DATA CARD WITH ID: $ HAS A SPANWISE INDEX (SPANID) = $ WHICH IS LESS THAN 1 OR GREATER THAN THE'
'NUMBER OF SPANWISE BOXES (NSPAN) = $.'
'A $ BULK DATA CARD WITH ID: $ DOES NOT HAVE COMPLETED X-CPU-CPL PAIRS (I.E. IN THREES).'
'A $ BULK DATA CARD WITH ID: $ HAS A X-LOCATION VALUE GREATER THAN 100 PERCENT CHORD.'
'A $ BULK DATA CARD WITH ID: $ HAS X-LOCATION VALUES THAT ARE NOT IN ASCENDING ORDER.'
'A $ BULK DATA CARD WITH ID: $ HAS CHORDWISE X-VALUES THAT ARE NOT IN ASCENDING ORDER,'
'SPECIFIED IN $ BULK DATA CARD WITH ID: $.'
'A $ BULK DATA CARD WITH ID: $ HAS CHORDWISE X-VALUES THAT DO NOT START WITH 0.0 OR END WITH 100.0'
'IN $ BULK DATA CARD WITH ID: $.'
'A $ BULK DATA CARD WITH ID: $ SPECIFIES $ (ITAX) NUMBER OF CHORDWISE HALF THICKNESS VALUES ($),'
'BUT ONLY $ ARE LISTED IN THE CORRESPONDING $ BULK DATA CARD WITH ID: $.'
'A $ BULK DATA CARD WITH ID: $ SPECIFIES $ (ITAX) NUMBER OF CHORDWISE CAMBER VALUES ($),'
'BUT ONLY $ ARE LISTED IN THE CORRESPONDING $ BULK DATA CARD WITH ID: $.'
'A $ BULK DATA CARD WITH WID: $ HAS $ NUMBER OF CHORDWISE DIVISIONS (NCHORD) SPECIFIED,'
'BUT ONLY $ VALUES ARE LISTED IN THE CORRESPONDING $ BULK DATA CARD WITH ID: $.'
'A $ BULK DATA CARD WITH ID: $ REFERENCED BY A $ BULK DATA CARD WITH ID: $'
'IS NOT DEFINED AS THE CENTERLINE OF THE BODY.'
'A $ WING MACROELEMENT WITH WID: $ HAS ZERO AREA.'
'DUPLICATED ID IN BULK DATA CARD $ WITH ID: $.'
'ERROR IN BULK DATA ENTRY $ WITH ID: $. NUMBER OF INLET PANELS EQUALS $ (INLET).'
'BUT THERE ARE $ NUMBER OF BOX ID SPECIFIED.'
```

```
*MODULE 36     ZONA'S SPLINZ MODULE MESSAGES
'$ ENTRY $ REFERENCES AN AERODYNAMIC BODY COMPONENT.  ONLY WING-LIKE COMPONENTS ALLOWED.'
'COORDINATE SYSTEM $, REFERENCED ON $ ENTRY $, CANNOT BE FOUND.'
'GRID POINT $, REFERENCED ON $ ENTRY $, CANNOT BE FOUND.'
'$ ENTRY $ REFERENCES STRUCTURAL SET DEFINITION $ THAT DOES NOT EXIST.'
'THE STRUCTURAL SET DEFINED BY SET2 ENTRY $, REFERENCED ON $ ENTRY $, IS EMPTY.'
'THE STRUCTURAL POINT DEFINITION PRISM DEFINED BY SET2 ENTRY $ ON $ ENTRY $ HAS ILLEGAL GEOMETY.'
'$ ENTRY $ RESULTS IN A SINGULAR TRANSFORMATION MATRIX.'
'AERODYNAMIC BOX WITH INTERNAL IDENTIFICATION NUMBER $ HAS BEEN SPLINED MORE THAN ONCE.'
'$ ANALYSES ARE REQUESTED IN SOLUTION CONTROL BUT NO SPLINE OR ATTACH ENTRIES EXIST.'
'NO COORDINATE SYSTEM FOR THE SPLINE Y-AXIS IS DEFINED ON $ ENTRY $.'
'  WHEN USED ON A LIFTING SURFACE A CID MUST BE SUPPLIED.'
'$ SETID $ SPECIFIES NON-EXISTENT MACRO-ELEMENT $.'
'$ SETID $ SPECIFIES NON-EXISTENT AERODYNAMIC BOXES FOR MACRO-ELEMENT $.'
'THE RECTANGULAR REGION SPECIFIED BY BOX1 AND BOX2 ON $ SETID $ CONTAINS NO AERODYNAMIC BOXES.'
'$ SETID $ SPECIFIES MORE BOXES THAN EXIST IN THE AERODYNAMIC MODEL.'
'$ SETID $ SPECIFIES DUPLICATE AERODYNAMIC BOXES MACROID $, EXTID $.'
```

```
'$ SETID $ SPECIFIES NON-EXISTENT AERODYNAMIC BOX MACROID $, EXTID $.'
'COORD SYS $, REFERENCED ON $ ENTRY, CANNOT BE FOUND.'
'$ $ SPECIFIES A SPLINE PLANE WHICH IS NEARLY PERPENDICULAR TO THE FREE STREAM VELOCITY.'
'$ SETID $ SPECIFIES AERODYNAMIC BOXES BELONGING TO MORE THAN ONE MACRO-ELEMENT.'
'$ SETID $ FAILS WHEN USING DEFAULT SPLINE PLANE (CP=BLANK) BECAUSE THE BOUNDARY FOR'
'   MACRO-ELEMENT $ DOES NOT DEFINE A PLANE.  USE CP OPTION TO SPECIFY A REFERENCE PLANE.'
'SPLINE2 WITH ID: $ CAN ONLY BE USED WITH CAERO7.'
'AERODYNAMIC GRID WITH INTERNAL ID: $ CANNOT BE FOUND IN ATTACH BULK DATA ENTRY.'
'STRUCTURAL GRID WITH EXTERNAL ID: $ CANNOT BE FOUND IN ATTACH BULK DATA ENTRY.'
'SPLINE2 WITH ID: $ HAS LESS THAN TWO STRUCTURAL GRIDS.'
'$ WITH ID: $ ERROR. STRUCTURAL GRID WITH INTERNAL ID: $ CANNOT BE FOUND.'
'SPLINE2 WITH ID: $ HAS TWO STRUCTURAL GRIDS WITH ID: $ AND $ THAT SHARE THE SAME '
'  LOCATION ALONG THE LINE OF THE SPLINE.'
'THE $ $ AERODYNAMIC BOX IS NOT ATTACHED TO THE STRUCTURE, THEREFORE, NO DISPLACEMENT'
'  IS ASSUMED FOR THIS BOX.'
'AERODYNAMIC GRID WITH INTERNAL ID: $ CANNOT BE FOUND.'
'SPLINE1 WITH ID: $ SPECIFIES A SPLINE PLANE WHICH IS NEARLY PERPENDICULAR TO THE FREE'
'   STREAM VELOCITY.'
'$ WITH ID: $ REFERS TO A SETi THAT HAS LESS THAN $ GRIDS.'
'$ WITH ID: $ REFERS TO A SETi THAT HAS ALL GRIDS ALIGNED ALONG A LINE.'
'$ WITH ID: $ REFERS TO A SETi THAT HAS TWO GRIDS AT THE SAME LOCATION.'
'$ WITH ID: $ GIVES A SINGULAR MATRIX.'
'A REFERENCED LOCAL COORDINATE SYSTEM WITH ID: $ CANNOT BE FOUND.'
'SPLINE3 WITH ID: $ REFERS TO A SETi THAT HAS ALL GRIDS LOCATED ON THE SAME PLANE.'
'  THE NORMAL VECTOR OF THE PLANE IS XN = $, YN = $, ZN = $ .'
```

```
*MODULE 37      ZONA'S ZAEROM MODULE MESSAGES
'$ WITH ID: $ HAS DUPLICATED REDUCED FREQUENCIES.'
'THERE IS NO CAERO7 OR BODY7 INPUT FOR THE ZAERO MODULE.'
'THERE IS NO OR MORE THAN ONE $ INPUT FOR THE ZAERO MODULE STEADY/UNSTEADY AERODYNAMIC ANALYSIS.'
'REFERENCE GRID ID FOR MOMENT CENTER (GREF = $) REFERENCED IN $ DOES NOT EXIST.'
'THE CONTROL POINT OF AN AERODYNAMIC BOX WITH ID: $ LOCATED ON A CAERO7 WING MACROELEMENT'
'  WITH WID: $ ALIGNS WITH THE EDGE OF ANOTHER AERODYNAMIC BOX WITH ID: $ LOCATED ON A'
'  CAERO7 WITH WID: $.'
'THE CONTROL POINT OF AN AERODYNAMIC BOX WITH INTERNAL ID: $ LOCATED ON A CAERO7 WITH INTERNAL'
'  ID: $ ALIGNS WITH THE EDGE OF ANOTHER AERODYNAMIC BOX WITH INTERNAL ID: $ LOCATED ON A'
'  CAERO7 WITH INTERNAL ID: $.'
```